GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TRABAJO FIN DE GRADO

SISTEMAS DE APRENDIZAJE AUTOMÁTICO PARA RECONOCIMIENTO DE PERSONAS MEDIANTE GAIT

Autor: Alberto Montes de Andrés
Tutor: Raúl Sánchez Reíllo

Leganés, 15 de febrero de 2019

Agradecimientos

Gracias a toda la gente que me ha ayudado a lo largo de este camino. A la gente que ha estado conmigo y ha podido verlo y a la gente que estuvo conmigo, pero ya no está.

Papá, mamá, Silvia y Estefi, un pedazo de este trabajo es vuestro.

Resumen

El trabajo que se va a presentar a lo largo del siguiente documento consistirá principalmente en poder comprobar si una base de datos dada puede ser analizada para distinguir a diferentes usuarios a través de mediciones de su pisada con un dispositivo móvil.

Dicha base de datos estará compuesta por las aceleraciones en los tres ejes y el tiempo que transcurre entre una muestra y otra. Cada usuario tendrá un número determinado de pruebas realizadas en distintos escenarios.

Para ello será necesario un preprocesamiento, tratamiento y adaptación de los datos a formatos específicos, lo cual llevará gran parte del desarrollo de la memoria para así después poder introducirlos en un programa de minería de datos llamado WEKA.

Con este programa se pretende, a través de la elección de distintos clasificadores basados en algoritmos de machine learning, clasificar todas las instancias de la base de datos y obtener el porcentaje de acierto o correspondencia con cada usuario.

Por último, cabe destacar la novedad de este trabajo, ya que trata acerca de un tema no muy estudiado en profundidad en la actualidad, lo cual deja muchas posibles opciones de estudios futuros.

Executive Summary

The following End-Of-Degree-Project that will be presented throughout this document will test whether a given database can be analyzed to distinguish different individuals through measurements of their gait with a mobile device.

This database will be composed of the accelerations in the three axes and the time between one sample and another. Each user will have a certain number of tests carried out in different scenarios.

This will require pre-processing, processing and adaptation of the data to specific formats, which will take much of the development of this document. Afterwards, all the data will be introduced into a data mining program which is called WEKA.

Through the selection of different classifiers based on machine learning algorithms, the aim of this program is to classify all the instances of the database and obtain the percentage of success or correspondence with each user.

Finally, it is worth mentioning the innovation of this work, since it deals with a subject that has not been studied in depth at present, which leaves many possible options for future studies.

Índice

Índice de Figuras

Índice de Tablas

Listado de Acrónimos

CLI	Command-Line Interface
DNI	Documento Nacional de Identidad
EEUU	Estados Unidos
FBI	Federal Bureau of Investigation
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
ML	Machine Learning
RGPD	Reglamento General de Protección de Datos
TFG	Trabajo Fin de Grado
TI	Tecnologías de la Información
UC3M	Universidad Carlos III de Madrid
URL	Uniform Resource Locator
WEKA	Waikato Environment for Knowledge Analysis

1 Introducción

Hoy en día se manejan grandes cantidades de información en todos los ámbitos de la sociedad, por lo que es necesaria la existencia de programas que analicen estos datos de manera efectiva y eficaz.

Es por eso que la idea principal sobre la que se va a desarrollar este Trabajo de Fin de Grado es el análisis de una base de datos, la cual ha sido recogida previamente por fuentes ajenas al trabajo para su posterior análisis. Esto se realizará mediante un programa de minería de datos llamado WEKA.

La base de datos constará de unos parámetros basados en tiempos y aceleraciones sobre los que gracias al programa anteriormente mencionado. A través de algoritmos de machine learning, tales como arboles de decisión, se conseguirá un aprendizaje automático para el reconocimiento de personas según los distintos parámetros de su pisada.

A lo largo de este trabajo se desarrollará un preprocesamiento de la base de datos para su adecuamiento hacia el programa y, más adelante, se realizará un análisis de dicha base de datos en la que se evaluarán sus resultados.

1.1 Motivación

El tema de este trabajo se sale un poco de las bases de la electrónica y la automática común y se desarrolla en una rama más cercana al campo de la informática. Es por esto que el hecho de realizar este trabajo nace de la curiosidad y el afán por ampliar el conocimiento en una rama de la ingeniería desconocida hasta el momento.

Es interesante también profundizar en este tema por dos razones: por un lado, las ideas principales sobre las que se desarrolla este trabajo son temas muy presentes en la actualidad; y, por otro lado, no existen muchos trabajos recientes acerca de este tema debido a que los estudios de la pisada en términos biométricos son algo muy novedoso.

La idea de realizar este Trabajo de Fin de Grado sobre un sistema de aprendizaje automático para el reconocimiento de personas mediante gait surge del interés de como a partir de algo tan simple como la pisada de una persona se puede llegar a reconocer a esta misma, con una simple recogida de muestras y un programa informático. Esto es gracias a los avances tecnológicos de hoy en día, ya que esto era impensable hace unos años.

1.2 Objetivos

El objetivo principal de este trabajo es conseguir el reconocimiento de personas a partir de una base de datos con distintos parámetros de la pisada. Todo ello se pretende alcanzar gracias al entrenamiento de los datos y su posterior análisis basado en algoritmos de machine learning.

Además de este objetivo general, el trabajo consta de una serie de objetivos específicos tales como:

- Preparación y procesamiento de las bases de datos incluyendo el uso de los lenguajes de programación necesarios.
- Uso de clasificadores, así como su posterior análisis y su impacto en las muestras de la base de datos.
- Usar un programa de minería de datos, en este caso WEKA, y entender sus funcionalidades.
- Conseguir un porcentaje de acierto en el reconocimiento de usuarios notablemente alto.

1.3 Estructura del documento

Este trabajo se encuentra dividido en distintos epígrafes que serán desarrollados a lo largo del documento. A continuación, se expondrá un breve resumen de cada capítulo del mismo para entender mejor el contenido de una manera global, así como, su estructura y las distintas etapas en las que está planteado.

Capítulo 1 – Introducción.

En este capítulo se introduce una breve explicación acerca de la motivación que ha llevado al desarrollo del trabajo, así como una breve descripción de los objetivos. Por otra parte, también se presenta un breve desarrollo del marco regulatorio del proyecto y su entorno socioeconómico. Por último, se concluye con un desglose de la estructura que tendrá este trabajo.

Capítulo 2 – Estado del arte.

Este capítulo expone el ámbito en el que se va a desarrollar el trabajo, ya que lo largo de este capítulo se ofrece una visión global de las distintas partes de la biometría. Se explicará una pequeña introducción junto con una breve reseña a su historia, para así, en los siguientes puntos hablar acerca de su evolución, etapas y técnicas. Por último, este punto se centrará en el análisis de la marcha (gait) para presentar y tratar sus diferentes aspectos.

Capítulo 3 – Plataforma de desarrollo.

En este capítulo se ofrece un amplio resumen acerca de la plataforma de desarrollo que ha sido usada en este trabajo. Se mostrarán tanto las principales partes de dicho programa, como una explicación para su posterior uso.

Capítulo 4 – Diseño de la solución.

En este capítulo se expondrá todo el diseño de la solución que tiene dicho Trabajo de Fin de Grado. No obstante, se explicará también el porqué de todas las decisiones tomadas a lo largo del trabajo, así como las otras soluciones pensadas posibles con sus correspondientes ventajas y desventajas.

Capítulo 5 – Desarrollo de la solución.

En este capítulo se explicará el cómo se ha llevado a cabo todo el contenido del trabajo paso a paso mediante breves capítulos que dividirán este punto de forma cronológica. Todo ello irá acompañado de las correspondientes imágenes e indicaciones que harán el proceso de desarrollo más fácil de entender.

Capítulo 6 – Problemas surgidos.

Este capítulo resumirá los problemas con los que se ha encontrado el alumno a lo largo del trabajo. En él, se podrán leer todas las complicaciones que se han presentado y las correspondientes soluciones tomadas.

Capítulo 7 – Conclusiones y líneas futuras.

En este capítulo se expondrán las conclusiones que el alumno ha sacado de la realización de este trabajo, tanto a nivel personal como a nivel técnico. También se podrán leer las líneas futuras de trabajo e investigación que pueden tener lugar acerca del tema expuesto en la memoria.

1.4 Marco regulatorio y entorno socioeconómico

En relación al marco regulatorio cabe destacar que, al ser un trabajo desarrollado con los parámetros de la pisada de distintos usuarios, todos los datos recogidos en la primera base de datos con la que se trabajará se encontrarán protegidos bajo el Reglamento General de Protección de Datos (RGPD). Este reglamento servirá para preservar la identidad de las personas cuyos datos se encuentren en la base de datos.

Por otro lado, en cuanto al entorno socioeconómico, lo primero que es necesario destacar es que no se ha necesitado ningún gasto monetario para componentes o materiales para el desarrollo de este trabajo. Únicamente se ha necesitado de programas de libre descarga para su desarrollo, lo que hace que este trabajo sea muy asequible tanto para particulares como para todo tipo de empresas.

Por último, este trabajo se enmarca dentro de áreas aún no desarrolladas por completo, es decir, aún no hay sistemas que puedan reconocer al usuario por la pisada de una manera totalmente fiable y real. Es por esto que, en un futuro, este trabajo podrá ser considerado de gran ayuda a la hora del reconocimiento de usuarios en materia de seguridad ciudadana en distintos posibles escenarios.

2 Estado del arte

2.1 Introducción a la biometría

La biometría [1] (del griego *bios* vida y *metron* medida) es la toma de medidas estandarizadas de los seres vivos o de procesos biológicos. Se llama también biometría al estudio para el reconocimiento inequívoco de personas basado en uno o más rasgos conductuales o físicos intrínsecos.

En las tecnologías de la información (TI), la «autentificación biométrica» o «biometría informática» es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo, para su autentificación, es decir, «verificar» su identidad.

Aunque los sistemas biométricos automatizados de hoy en día están basados en ideas que fueron concebidas hace cientos o miles de años, no ha sido posible llevarlos a la práctica hasta hace unas décadas debido a los grandes avances tecnológicos que suponen.

Desde los comienzos de la civilización, [2] los seres humanos han tomado el rostro como la principal característica para identificar a sus conocidos y también a desconocidos. Además del rostro, otras características usadas de manera inconsciente para el reconocimiento de individuos son el habla, la voz o la marcha (gait).

2.1.1 Historia de la biometría

La primera constancia del uso de la biometría [2] es en el siglo XIV en China cuando los mercaderes usaban las palmas de las manos y de los pies untadas en tinta y plasmadas en un

papel para reconocer a los niños. Algunas fuentes aseguran que incluso los egipcios ya empleaban la biometría para la identificación de personas a través de su firma.

Sin embargo, la práctica de la biometría no comenzó a utilizarse en occidente hasta el siglo XIX. Concretamente en 1858 Sir William Herschel en la India plasmó las huellas de las palmas de las manos por detrás de cada contrato de sus trabajadores para así distinguirlos llegado el día en que cobraran.

Unos años más tarde, Alphonse Bertillon desarrolló un método para identificar personas que más tarde se conocería como ‘‘Bertillonage’’. Este método se basaba en la recopilación de medidas del cuerpo, descripciones físicas y fotografías de cada individuo. Sin embargo, en 1903 se acabó descartando ya que descubrieron que algunas personas poseían las mismas medidas, concretamente se dio un caso en el que dos gemelos fueron condenados por tener prácticamente las mismas medidas.

No fue hasta 1892 cuando el antropólogo inglés Sir Francis Galton desarrolló un estudio para un nuevo tipo de identificación y clasificación de las personas en el que recogió puntos característicos de las huellas dactilares de los diez dedos de las manos. Actualmente, se sigue empleando este sistema.

En 1896, Sir Edward Henry y su ayudante de policía de la India Azizul Haque, consultaron con Galton sobre la recogida de las huellas dactilares para la identificación de delincuentes. Más adelante, cuando se empezó a emplear este sistema, desarrollaron un método para almacenar y buscar información ya recogida. A este sistema se le conoció como Sistema de Clasificación Henry, el cual fue utilizado durante un tiempo por el FBI y otras organizaciones de justicia penal. Unos años después incluso se llegó a crear la Oficina de Huellas Dactilares de New Scotland Yard donde se empleaba este sistema.

En 1969 el FBI empezó a plantear un sistema para automatizar el reconocimiento y clasificación de personas por sus huellas dactilares ya que habían llegado a un punto de sobrecarga de huellas para hacerlo manualmente. Fue años más tarde cuando desarrollaron una tecnología para la extracción de puntos característicos junto con su correspondiente lector. Esto suponía un alto coste de almacenamiento digital, por lo que se desarrollaron métodos de digitalización de huellas y sistemas de compresión sobre la calidad de las imágenes y demás propiedades. Todo ello fue conseguido gracias a la creación del algoritmo M40 utilizado para estrechar la búsqueda humana y crear un conjunto más pequeño de imágenes.

En las siguientes décadas se fueron viendo numerosos avances en los diferentes sistemas de identificación de personas tales como el uso de patrones de reconocimiento del iris, el reconocimiento facial y del habla.

En 1988 se desarrolló la técnica conocida como Eigenface que estaba basada en el álgebra lineal para el reconocimiento facial. Ya en la década de los 90, Turk y Pentland descubrieron que el error residual de esta técnica podía ser usado para dar un paso hacia el reconocimiento facial en tiempo real. Además, unos años después, se patentó el primer algoritmo de reconocimiento del iris y la geometría de las manos se llegó a aplicar en los Juegos Olímpicos de Atlanta ’96.

Entrando en el siglo XXI, el gobierno de los EEUU, el FBI y el Departamento de Defensa comenzaron a incluir datos de iris, cara, palma y huellas dactilares en las bases de datos de los

registros del país para así emplear la biometría para la identificación de criminales e intentar evitar posibles ataques terroristas.

A día de hoy, todos los dispositivos electrónicos de bolsillo cuentan con un lector de huella digital para el reconocimiento de su usuario y dotándoles, de esta manera, de una mayor seguridad. El pionero en esta técnica fue Apple en el año 2013 al incluirlo en su último modelo de smartphone.

2.1.2 Evolución de la biometría

Dados los límites existentes en la sociedad en cuanto al reconocimiento de personas y su importancia en los ámbitos en la actualidad, surge la necesidad de tomar otras vías para ello. De aquí nace la biometría, es decir, el reconocimiento e identificación de personas a partir de sus rasgos fisiológicos y biológicos.

En la actualidad, el auge de las comunicaciones y avances tecnológicos que permiten, entre otras cosas, operar a grandes distancias, supone una mayor complejidad para que el reconocimiento de la identidad de las personas se realice de una manera más segura.

Antes de la existencia de la biometría, la manera de distinguir a las personas era únicamente mediante documentos de identificación, como puede ser el DNI o el pasaporte, o bien por contraseñas. Hoy en día, gracias a las nuevas tecnologías, estas formas de identificación han ido avanzando y mejorando, siendo ya mucho más complejas y seguras, para evitar así posibles manipulaciones. Es por esto que surge la biometría, para aumentar esta seguridad, ya que no es igual duplicar un documento de identificación, que una huella dactilar o un escáner de retina.

La biometría es utilizada en diferentes sectores con un amplio abanico de funciones, siendo alguna de las más corrientes la verificación de usuarios para las aplicaciones móviles, la recogida de huellas dactilares con fines policiales o incluso el uso de transferencias e importantes movimientos bancarios. Todos estos tipos de reconocimiento y otros más serán explicados en profundidad a lo largo del trabajo.

2.1.3 Etapas del reconocimiento biométrico

Muchas de las características físicas y biológicas de las personas pueden llegar a ser estudiadas y usadas como elemento para la identificación biométrica, por lo que existen muchas técnicas diferentes. Sin embargo, todas estas técnicas de identificación biométricas tienen en común un esquema con una correlación de fases independiente a la técnica utilizada.

Los sistemas de identificación que se van a desarrollar en este apartado se basan en dos esquemas totalmente diferentes: [3]

1. **Reclutamiento**: en primer lugar, es necesaria la recogida de muestras del individuo a estudiar, las cuales serán procesadas para más tarde extraer un patrón común a todas ellas y almacenarlo así.

   Si en el conjunto de datos obtenido es necesaria la toma de más de una muestra o la toma de muestras de varios individuos, normalmente se suele hacer la media de las características obtenidas de las muestras de cada individuo.

   En este proceso, una persona será encargada de la recogida de los datos, así como de supervisar y verificar que todo transcurra con normalidad.

2. **Utilización**: con el patrón ya extraído y almacenado, el individuo podrá utilizar el sistema, el cual comparará sus características con las del patrón almacenado viendo de esta manera el porcentaje de error o acierto en la comparación.

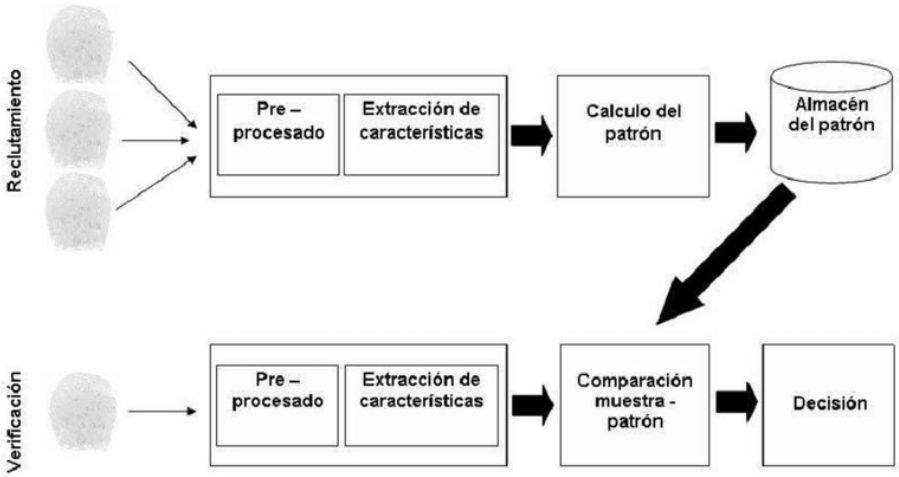

Figura 1. Etapas del reconocimiento biométrico.

Según se puede observar en la imagen, el reclutamiento y la utilización mencionados anteriormente constan de las siguientes fases:

- **Captura**: en esta fase la recogida de los datos físicos, biológicos o de comportamiento del usuario se puede realizar de varias formas, dependiendo principalmente, de la técnica biométrica empleada. A veces, incluso siguiendo con la misma técnica biométrica, la recolección de muestras del usuario puede realizarse de diferentes maneras. Así, por ejemplo, una huella dactilar puede tomarse mediante cámara de vídeo

o ultrasonidos y la dinámica de la marcha (gait) puede tomarse por sistemas de percepción visual o mediante acelerómetros y giroscopios.

- **Preprocesado**: dependiendo de la técnica utilizada, esta fase se ocupa principalmente de adaptar los datos para que posteriormente se pueda realizar una extracción de las características de éstos de manera que contengan los menores errores posibles. Un ejemplo de preprocesado sería quitar el ruido de una imagen, eliminando datos fuera de rango o simplemente corrigiendo el contraste de una imagen para hacer así, su posterior extracción de características más sencilla.
- **Extracción de características**: dentro de todas las fases, esta es la más importante ya que es en la que se establece la capacidad de distinguir entre individuos. En este bloque se pueden tomar direcciones muy distintas, aun estando dentro de una misma técnica. También es común usar técnicas basadas en redes neuronales como entrenamiento de los datos para posteriormente obtener mejores resultados.
- **Comparación**: las características de las muestras se comparan con el patrón ya almacenado. Las variaciones de las muestras, bien sea por variaciones en la recogida de muestras o por variaciones en las características, dan como resultado una probabilidad de semejanza. Esta comparación puede realizarse con diferentes métodos como Métricos (Distancia Euclídea), Estadísticos (funciones de distribución) o técnicas de modelado de problemas (GMM).

  Para establecer el porcentaje de acierto y error de la comparación, se establece un determinado umbral. El nivel que se ha de escoger para dicho umbral dependerá de la seguridad que se quiere llegar a conseguir; no podrá ser ni demasiado alto, ya que provocaría una menor restricción, ni demasiado bajo porque provocaría un falso rechazo en las muestras.

Hasta el momento se ha desarrollado la identificación biométrica desde un punto de vista teórico, sin embargo, ésta puede basarse y apoyarse en dos diferentes esquemas de funcionamiento:

- **Reconocimiento**: su principal tarea es la de encontrar a un individuo comparando sus características, obtenidas anteriormente con los patrones, entre todas las que se encuentran almacenadas en una base de datos del sistema. El resultado de acierto de la comparación dependerá de si el sistema consigue que la mayor probabilidad supere el umbral e identifique así al sujeto.
- **Autenticación o verificación:** el sistema intenta averiguar si el individuo es quien dice ser. Primero el sujeto confirma su identidad al sistema, y éste compara las características del sujeto en cuestión con el patrón ya almacenado en la base de datos. Quedará en manos del umbral si la verificación de este usuario es aceptada (si lo supera) o si es denegada o errónea (no supera el umbral).

2.1.4 Técnicas de reconocimiento biométrico

A pesar de que la huella dactilar es la característica más empleada para los sistemas de reconocimiento biométricos, hay otras muchas características físicas y biológicas de las personas que pueden ser susceptibles de ser usadas en la identificación biométrica. Algunas de las más extendidas son las siguientes: [3]

- **Voz**: es una de las técnicas con mayores expectativas en el mercado global, en la que participa activa y principalmente el sector telefónico. Esta técnica se lleva estudiando y perfeccionando decenas de años, pero hasta el día de hoy no se han llegado a comprender del todo cuáles son los factores que dificultan que la voz sea identificable. Dentro de estos factores se encuentran desde una enfermedad que afecte a la voz, la edad o simplemente el canal por el que sea transmitida.
- **Huella dactilar:** como mencionado anteriormente, es la característica más usada en la biometría, ya que fue la primera técnica en emplearse hace cientos de años. Hay una serie de estudios que aseguran la estabilidad de la huella dactilar con el tiempo y la edad. La recogida de datos sobre las huellas dactilares se puede realizar de muchas maneras diferentes gracias a las nuevas tecnologías desarrollas hoy en día. En cuanto a la extracción de características de las huellas cabe destacar tres técnicas: la correlación de imágenes, la comparación de los surcos de la huella y la comparación de los poros del dedo.
- **Rostro**: los humanos utilizan siempre esta técnica de manera inconsciente. A día de hoy, hay numerosos estudios que se están llevando a cabo para intentar lograr que esta técnica se acerque al nivel del resto. Se ha llegado a la conclusión de que esta técnica tiene un gran inconveniente, siendo este el efecto del tiempo sobre el propio rostro, ya que pueden cambiar diferentes características como la longitud del pelo, la barba, las gafas…
- **Iris**: esta técnica data de la década de principios de los 90 y es, a día de hoy, una de las técnicas más fiables y seguras que existen. Se basa principalmente en extraer características de la textura del iris ocular, ya que este es inalterable a la edad, y cada persona tiene siempre el mismo patrón, sin importar ningún factor externo. Debido a este dato, que hace de esta técnica única, se ha alcanzado un nivel de fiabilidad altísimo, por encima incluso de la huella dactilar. A pesar de todas estas ventajas cuenta con un inconveniente importante, es su elevado coste.
- **Marcha (gait)**: es una de las técnicas más novedosas y recientes, tanto es así que está aún en pruebas y desarrollándose. Se basa en una característica del comportamiento de las personas, por lo que puede ser falseada por la manipulación y otros factores externos.

Este trabajo gira en torno a la técnica de la marcha, por lo que será desarrollada con mayor profundidad en los siguientes apartados. [4]

Tabla 1. Ventajas y desventajas de las diferentes técnicas del reconomiento biométrico.

TÉCNICA	VENTAJAS	DESVENTAJAS
Voz	- Bajo coste - No necesidad de contacto - Sensores disponibles habitualmente	- Bajo rendimiento - Poca estabilidad - Baja unicidad
Huella dactilar	- Fiabilidad - Seguridad - Unicidad - Facilidad de uso	- Connotaciones policiales para el usuario
Rostro	- Inapreciable al usuario - No necesidad de contacto - Sensores disponibles habitualmente	- Baja unicidad - Poca estabilidad - Sensible ante cambios del usuario
Iris	- Alta unicidad - Alta estabilidad - Alta fiabilidad	- Alto coste - Incomodidad para el usuario - Difícil captura de los datos
Marcha (gait)	- Multitud de usos posibles	- Difícil recogida de datos - Poca fiabilidad por el momento

Todas estas técnicas tienen sus puntos favorables y sus puntos en contra, ventajas e inconvenientes que hacen que técnicas que dan resultados muy prometedores no puedan utilizarse debido a factores externos, como, por ejemplo, el rechazo del propio sujeto. Por esto, no se puede afirmar que exista una técnica idónea para la identificación biométrica que sea perfecta y se pueda emplear en el 100% de los casos.

2.1.5 Análisis de la marcha por la pisada (Gait)

El estudio de la marcha [5] ha ido evolucionando desde la primera vez que se analizó hasta el día de hoy de la mano de las nuevas tecnologías. En este tiempo, los métodos empleados para su análisis también han ido cambiando de tal forma que hoy en día facilitan estudiar los parámetros e identificar los factores que influyen en el patrón de la marcha de un modo más objetivo. Estos factores que afectan al análisis de la marcha pueden ser intrínsecos, como la edad, el género, la altura o la personalidad, o extrínsecos, como el tipo de terreno, el calzado o el transporte de carga.

2.1.5.1 Parámetros

El análisis de la marcha está constituido por distintos tipos de parámetros que pueden variar entre distintos individuos o incluso en un mismo individuo.

Cuando los resultados de estos parámetros se mantienen relativamente constantes, facilitan los datos del proceso del análisis de la marcha y mostrando así la relación e incidencia de los mismos desde un punto de vista de la identificación. [6]

Los parámetros más representativos son los siguientes:

- **Parámetros espaciales:** [7]
  - Ángulo de paso: indica el ángulo que forman el eje longitudinal del pie con la línea imaginaria de la dirección que sigue la marcha.
  - Longitud de paso: distancia en línea recta entre el talón de un pie con el talón del contrario.
  - Longitud de zancada: distancia en línea recta entre el talón del primer pie a apoyar con el talón del mismo pie cuando vuelve a ser apoyado.

- **Parámetros temporales:**
  - Apoyo: periodo de tiempo del total de la marcha en el cual el cuerpo permanece apoyado sobre un solo pie.
  - Balanceo: periodo de tiempo del total de la marcha en el que un pie permanece en el aire entre dos apoyos.
  - Doble apoyo: periodo de tiempo del total de la marcha en el que los dos pies se encuentran apoyados en el suelo.
  - Periodo de zancada: periodo de tiempo del total de la marcha en el que ocurren dos eventos iguales seguidos en un mismo pie.

- Periodo de apoyo: periodo de tiempo del total de la marcha en el que empieza el contacto de un pie con el suelo hasta que se despega por completo.
- Periodo de balanceo: periodo de tiempo del total de la marcha en el que un pie permanece en el aire entre dos apoyos.
- Cadencia: número de pasos por unidad de tiempo.

- **Parámetros espaciotemporales:**
  - Velocidad: espacio recorrido en la dirección de la marcha por unidad de tiempo.
  - Velocidad de balanceo: espacio recorrido por un pie cuando permanece en el aire entre dos apoyos por unidad de tiempo.

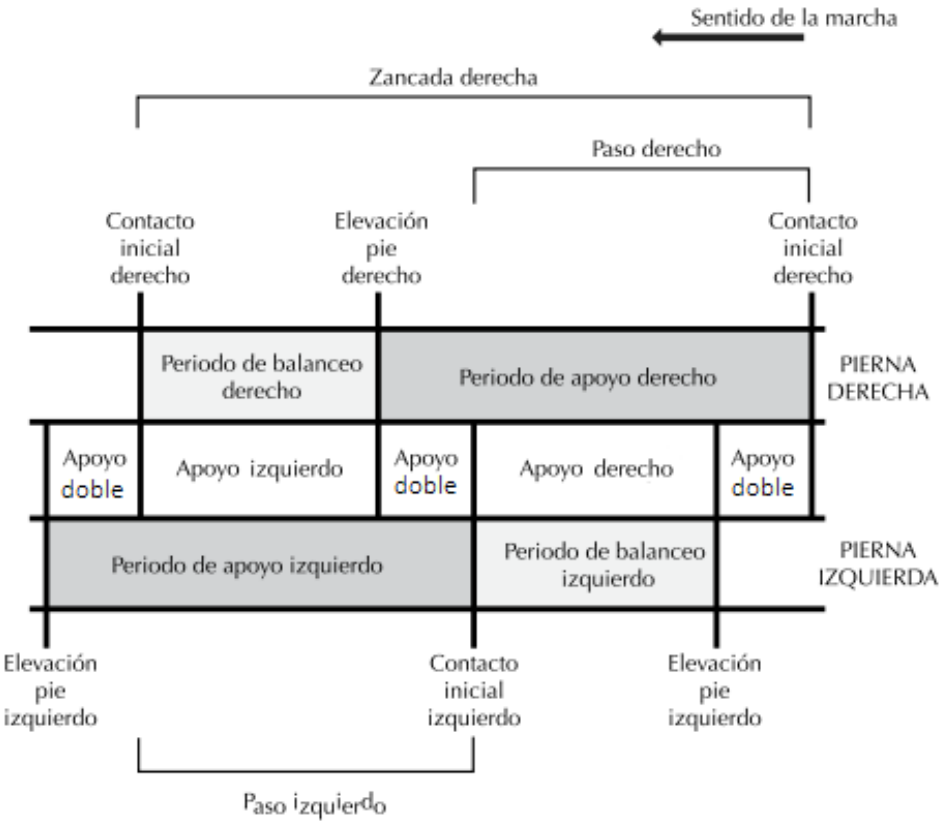

Figura 2. Representación gráfica del ciclo de la marcha

2.1.5.2 Técnicas

El estudio y análisis de la marcha consiste en identificar los distintos parámetros de una persona a través de la misma, consiguiendo así su reconocimiento. A continuación, se desarrollan las diferentes técnicas y campos en los que se ve involucrado este estudio: [8]

1. **Temporal o espacial:** en este campo la técnica más utilizada consiste en la medición de los diferentes parámetros que posee la marcha mediante un simple cronómetro o unas marcas en el suelo. Una vez medidos estos parámetros (paso, zancada, velocidad, cadencia…), se calculan las diferentes relaciones entre ellos para su posterior estudio.
2. **Cinemática:** el análisis de la marcha también se puede desarrollar mediante técnicas de medición del movimiento del cuerpo en el espacio. Hay varias técnicas dentro del campo de la cinemática, entre las que destacan:
    - Cronofotografía: consiste en la grabación del movimiento de la marcha utilizando luces estroboscópicas en una frecuencia concreta.
    - Grabaciones de vídeo: este método permite pasar de 2 dimensiones a 3 dimensiones por la obtención de distintos ángulos de las articulaciones y velocidades mediante una o más cámaras de vídeo. Todo esto es posible debido al desarrollo del software de análisis actual.
    - Sistemas de marcadores pasivos: se trata del uso de unos marcadores reflectantes colocados sobre el individuo a analizar para identificar sus movimientos. Mediante varias cámaras que emiten señales próximas a niveles infrarrojos combinadas con filtros capaces de detectar la reflexión procedente de los marcadores, se obtiene una triangulación basada en la diferencia de retardo entre la señal emisora y la receptora. Para que esto sea posible, existe un software que utiliza los parámetros de la triangulación para crear trayectorias en 3 dimensiones.
    - Sistemas de marcadores activos: se trata de una técnica muy similar a la anterior, pero con marcadores activos. En este caso no se produce la reflexión de una señal de infrarrojos ya que los marcadores emiten su propia señal con su respectiva frecuencia. La principal ventaja de esta técnica es que no es necesario procesar la señal para ubicar los marcadores.
    - Sistemas inerciales: estos sistemas, al contrario que los anteriores, no utilizan cámaras y están basados, como su propio nombre indica, en sensores inerciales.
3. **Cinética**: este campo trata acerca de la medición de las fuerzas que participan en la producción de los movimientos del cuerpo del individuo.
4. **Electromiografía dinámica**: se centra principalmente en el estudio de los patrones de actividad muscular durante la marcha. Dentro de los métodos que existen en este campo, el más común es la electromiografía de superficie, que consiste en una serie de

sensores que miden los impulsos musculares. Estos impulsos serán utilizados más tarde para reconocer los ciclos de la marcha.

2.1.5.3 Aplicaciones

A través del análisis de la marcha se analiza la capacidad de andar de cada individuo. Esto se consigue con una tecnología empleada principalmente en dos aplicaciones en la sociedad actual:[8]

- **Diagnóstico médico:** esta aplicación del análisis de la marcha puede estudiar cuáles son los factores que afectan a la capacidad de andar de las personas. Dicho análisis resulta de gran utilidad ya que puede mostrar síntomas de una enfermedad o bien reflejar la propia enfermedad. Por ejemplo, los pacientes que han sufrido una parálisis cerebral son las personas seleccionadas más habitualmente para el análisis de la marcha. En sus casos es posible diagnosticar y planificar ciertas estrategias a seguir para futuras intervenciones o sesiones de rehabilitación.
- **Identificación biométrica**: este documento se centra principalmente en el desarrollo de esta aplicación en concreto. Se utiliza principalmente para la identificación de personas a través de su forma de caminar. Para ello se estudian ciertos parámetros, algunos de los cuales han sido desarrollados previamente en el trabajo, relacionados entre sí que sirven para crear patrones que serán utilizados para el reconocimiento de personas. Como ejemplo de esto estaría la relación entre la altura y el paso de una persona.

  Esta aplicación también es usada en otros ámbitos como la autenticación de dispositivos electrónicos portátiles.

3 Plataforma de desarrollo

3.1 Introducción

La plataforma de desarrollo en la que se va a apoyar principalmente este trabajo va a ser un programa de aprendizaje automático y minería de datos conocido como WEKA. Permitirá abordar los desafíos más comunes de minería de datos, así como métodos relacionados con el aprendizaje automático. [9]

Este programa está diseñado para proporcionar una solución de trabajo en el aprendizaje automático a conjuntos de datos mediante un extenso listado de métodos y algoritmos.

Entre las actividades que se pueden destacar de WEKA, se encuentran algunas como:

- Ayudas al proceso de extracción de características de un conjunto de datos dado por el usuario.
- Preparación y preprocesamiento de un conjunto de datos de entrada para su posterior análisis.
- Valorar los resultados estadísticos de las muestras de datos y hacer una evaluación de estos.
- Visualizar los resultados y los conjuntos de datos. También se podrán visualizar todos los atributos de los datos y los errores en una posible clasificación.

WEKA es un programa sencillo que, a pesar de contener una multitud de herramientas para realizar todas estas actividades y muchas más, no requiere la escritura de ninguna línea de código. Por ejemplo, se puede preprocesar un conjunto de datos, establecer para él un sistema de aprendizaje y a través de un clasificador, analizar y evaluar sus resultados.

Para que estas actividades se puedan llevar a cabo, el programa necesita de unos datos de entrada introducidos previamente por el usuario. Estos datos pueden ser incluidos mediante bases de datos o simples ficheros.

En lo referente al diseño de la plataforma, la interfaz principal de WEKA se compone a su vez de distintas interfaces que tendrán distintos motivos o aplicaciones.

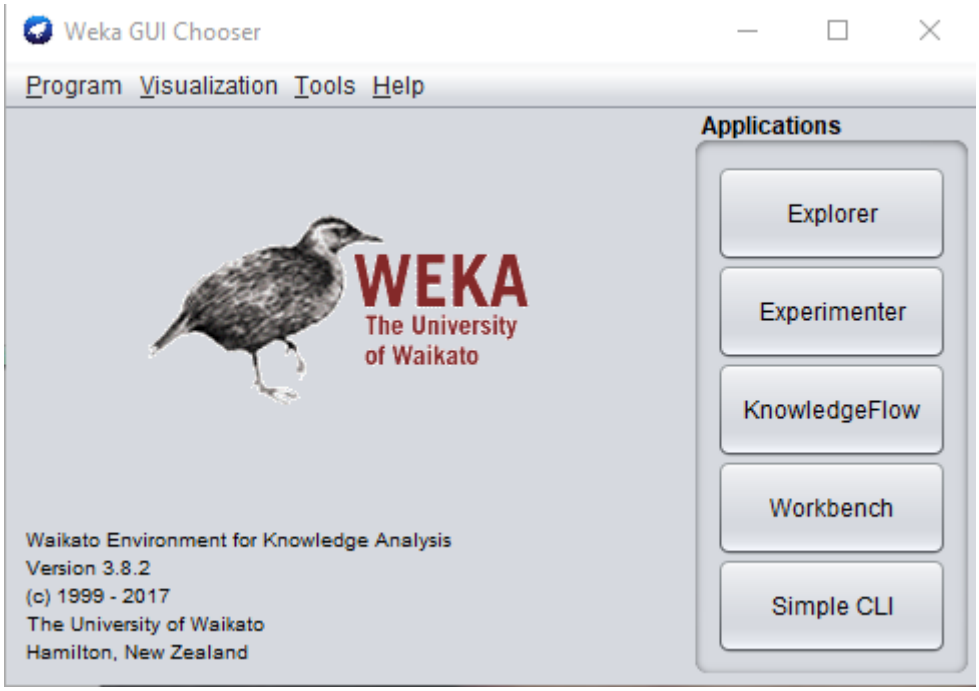

Figura 3. Interfaz principal de WEKA.

Como se puede observar, las distintas opciones que aparecen nada más abrir el programa son las siguientes: [10]

- **Explorer**: se trata de la interfaz principal de WEKA y en ella se podrá preprocesar, analizar, seleccionar y visualizar tanto los datos como los resultados obtenidos. En el caso de este trabajo es la más importante, ya que es en la que se va a centrar principalmente todo el desarrollo.
- **Experimenter**: esta segunda aplicación del programa tiene un uso, como su propio nombre indica, más experimental. En ella se pueden hacer ensayos a gran escala, de larga duración, por lo que se puede dejar analizando y volver para examinar los resultados.
- **Knowledgeflow**: es una interfaz parecida al *explorer*, pero con ligeras diferencias. En este caso se pueden diseñar y configurar el procesamiento de los datos en tiempo real. Proporciona también una alternativa al poder ver el flujo de datos en el sistema.
- **Workbench**: banco de trabajo exactamente igual que el explorar a excepción de alguna herramienta de personalización a la hora de trabajar.
- **Simple CLI**: es el acrónimo de interfaz simple de línea de comandos. Se trata de una consola que ofrece la opción de utilizar todas las opciones de WEKA a través de una línea de comandos del ordenador.

3.2 Interfaz *explorer*

Como se ha explicado antes, el explorer va a constituir la parte principal sobre la que se va a desarrollar este trabajo, por lo que es necesario explicar su funcionamiento y las principales opciones que contiene.

La ventana del *explorer* se compone de una serie de pestañas o paneles que dan acceso a las principales funcionalidades del programa. Algunas de las más importantes y las más utilizadas en este programa son las siguientes:

- Explorer.
- Classify.

3.2.1 Función *preprocess*

Esta pestaña se compone de una serie de ventanas más reducidas con diversas funcionalidades.

En esta ventana es donde se recogen los datos, los cuales se podrán cargar de distintas maneras, a saber:

- Mediante un archivo de texto compatible con el programa. En el caso de este trabajo se elegirá la opción de un archivo con extensión .arff y para lo que se empleará la opción que se encuentra arriba a la izquierda llamada *Open file*.
- Mediante una URL. El programa lee los datos de una URL y los procesa e introduce en el programa.
- Mediante una base de datos. Esta tiene que tener un formato compatible para la lectura del programa.

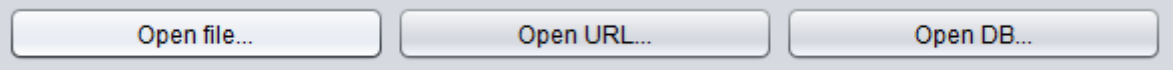

Figura 4. Opciones para cargar los datos.

Una vez importado el archivo como un conjunto de datos de entrada, se procede a explicar los distintos paneles de la ventana *preprocess*.

En primer lugar, se encuentra la ventana de *current relation*, que indica las principales características que tiene el archivo subido al programa. Estas características son las siguientes:

- **Relation**: nombre del archivo o base de datos.
- **Instances**: instancias que tiene el archivo, es decir, número de muestras distintas que tiene el conjunto de datos.
- **Atributes**: número de atributos que posee el archivo.
- **Sum of weights**: suma de los pesos asignados a cada instancia. En este caso será la unidad por el número de instancias.

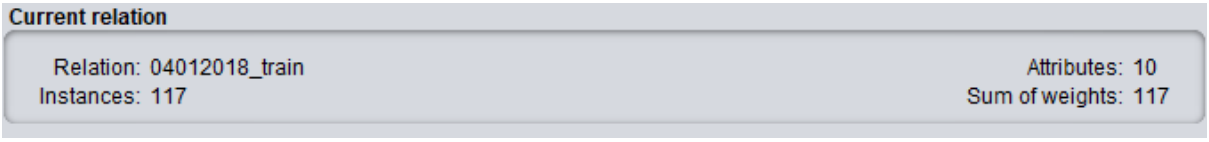

Figura 5. Ventana *current relation*.

La segunda ventana que se expone brevemente se corresponde a los atributos que poseerá el archivo. En principio no hay número limitado de atributos que puedan contenerse en la base de datos introducida.

Los atributos anteriormente referidos, aparecerán en el mismo orden que han sido listados en el archivo donde han sido declarados previamente. Junto a ellos el programa nos muestra cuatro opciones en forma de celdas para el manejo de los atributos:

- **All**: selecciona todos los atributos del desplegable.
- **None**: deselecciona todos los atributos del desplegable.
- **Invert**: activa todos los atributos deseleccionados y viceversa.
- **Pattern**: selecciona los atributos que coincidan con un patrón de búsqueda introducido por el usuario. Esta opción es útil en caso de que haya muchos atributos. De esta manera no será necesario buscar en la lista de todos los atributos para activarlos.

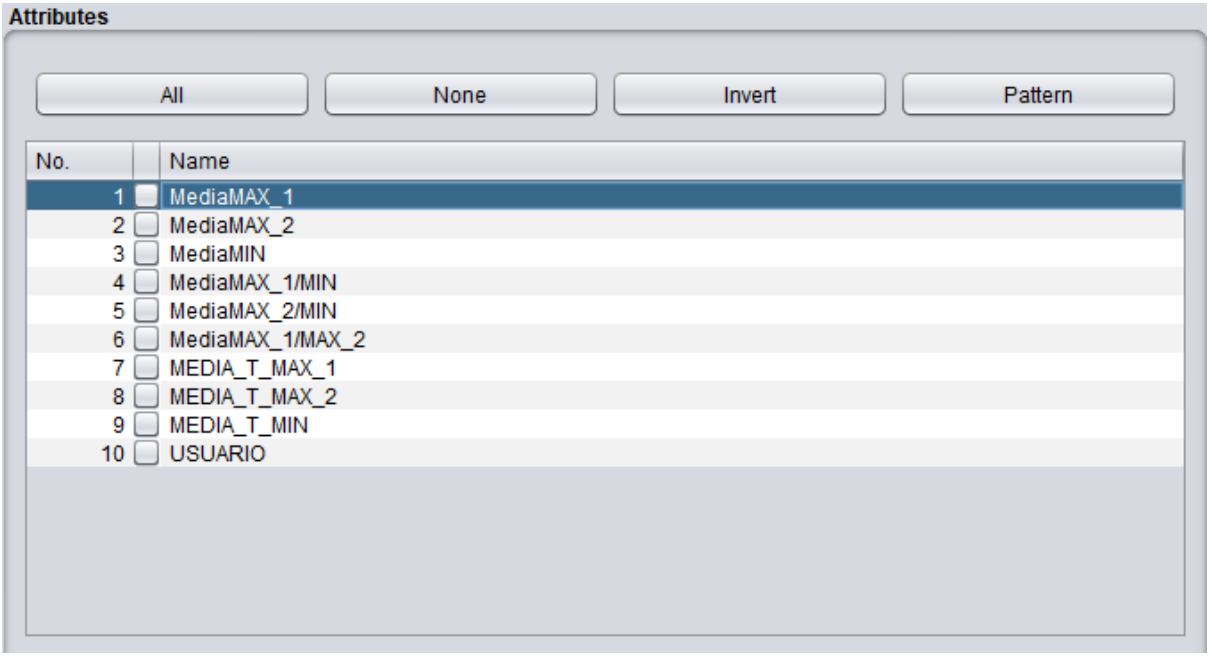

Figura 6. Ventana *attributes*.

Por último, se encuentra la tercera ventana llamada *selected attribute*, que se dividirá en 3 partes diferentes.

La primera de ellas consta de información sobre el atributo seleccionado, esto es, principalmente el nombre y el tipo de dato (numeric, flotante, string…).

Acto seguido, justo debajo, aparece la segunda parte dentro la ventana principal. En ella se muestran datos estadísticos del atributo previamente seleccionado, tales como el mínimo, el máximo, la media, y a desviación estándar.

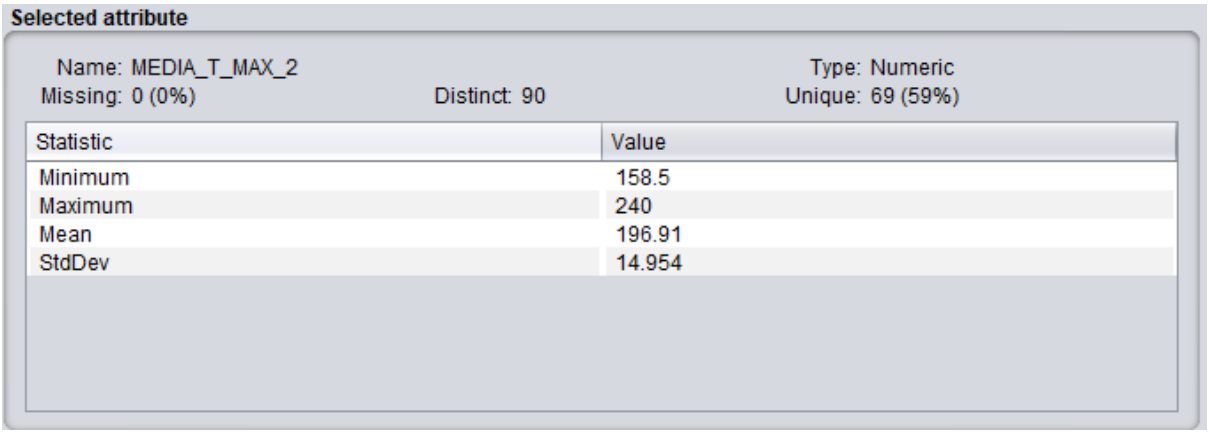

Figura 7. Ventana *selected attribute*.

Para terminar con esta ventana, la última parte consta de un panel de representación en el cual se pueden observar los atributos seleccionados. Posee un desplegable para visualizar el atributo que se desee en forma de grafico de barras y basado en los valores estadísticos previamente descritos.

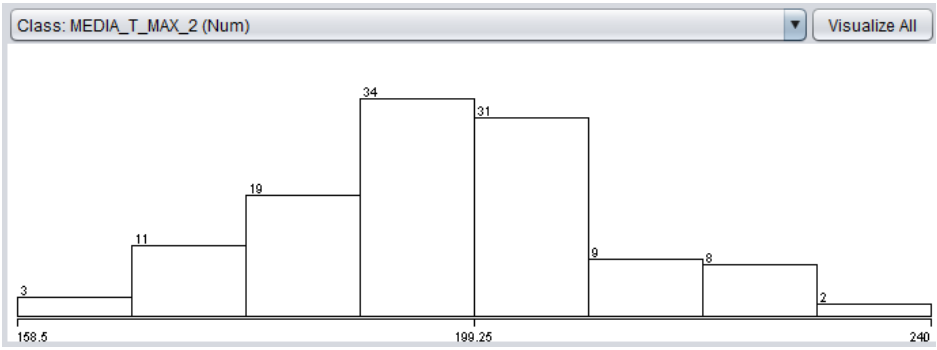

Figura 8. Visualización gráfica de los valores estadísticos.

Cabe destacar que también se puede usar la opción de *visualize all* como se puede comprobar en la anterior imagen, que muestra todos los atributos en su conjunto en una ventana emergente nueva.

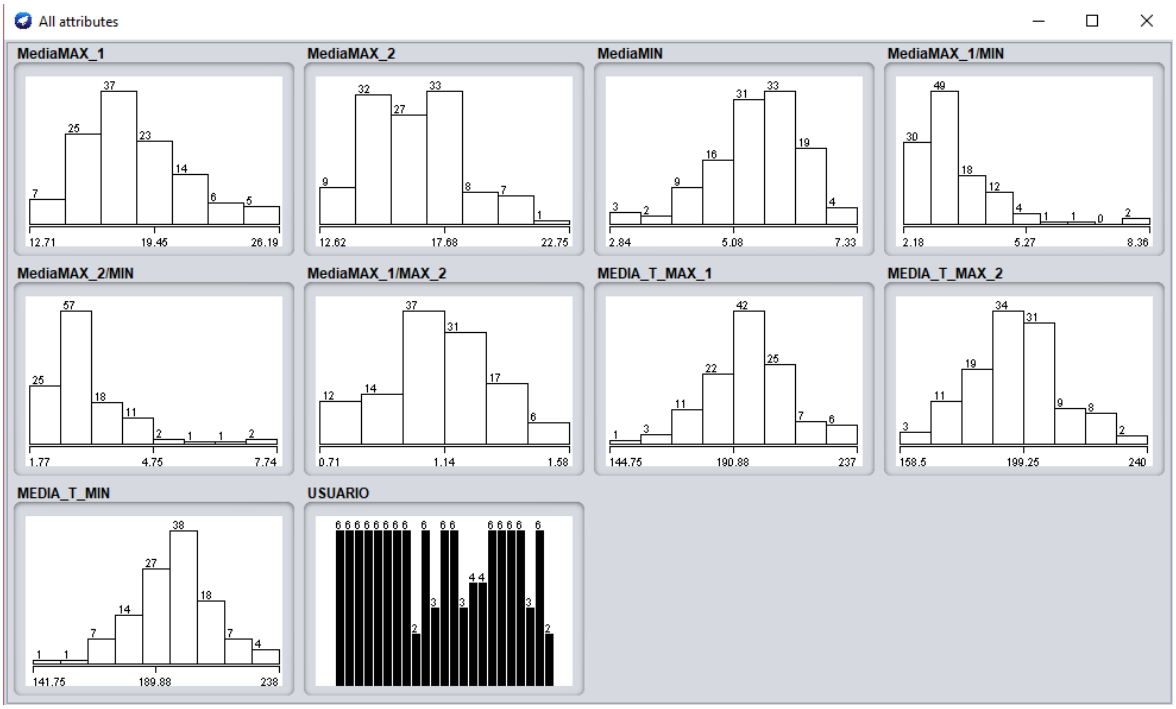

Figura 9. Visualización gráfica de todos los atributos.

3.2.2 Función *classify*

Junto con el *preprocess*, esta funcionalidad del explorer es una de las más importantes y las más usadas dentro de WEKA y de este trabajo.

Esta función ayuda a clasificar, como su propio nombre indica, las instancias de los datos mediante técnicas de clasificación y regresión. También se conoce como aprendizaje supervisado.

Este modo de clasificación [11] será recurrido siempre que se necesite encontrar un patrón de comportamiento en los datos de entrada del programa, para así encontrar relaciones entre atributos que ayuden a determinar un porcentaje de acierto y error en el análisis del estudio.

Todo ello se podrá conseguir mediante una serie de clasificadores pertenecientes a la biblioteca de WEKA, que más tarde serán explicados.

A su vez, esta pestaña de *classify* se compone, como en el anterior caso con la función *preprocess*, de varias secciones bien diferenciadas.

Nada más entrar en la función *classify*, lo primero que aparece es la ventana *classifier*, un menú desplegable en el que se podrá seleccionar el clasificador que se quiera utilizar para analizar los datos.

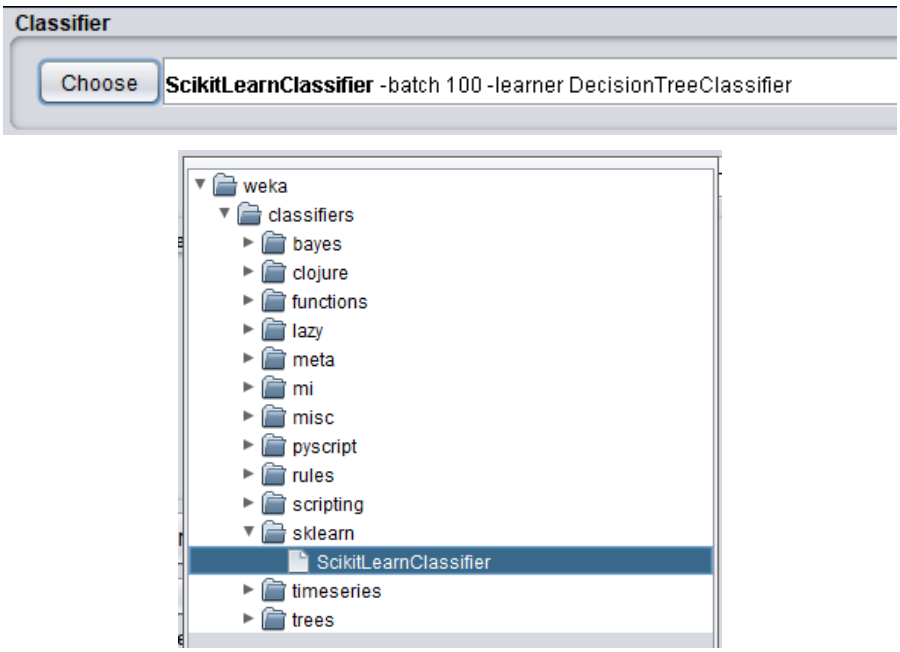

Figura 10. Ventana *classifier*.

Algunos de los grupos más importantes en los que se dividen los clasificadores son los siguientes: [11]

- **Bayes**: son métodos o algoritmos basados en el aprendizaje bayesiano. Representan las relaciones probabilísticas entre los atributos para encontrar entre todas las hipótesis la más probable. Entre algunos clasificadores destacados en la librería se encuentran NaivesBayes o HMM basado en el modelo de Markov.
- **Funciones**: en este grupo se aglutinan los métodos correspondientes a distintos tipos de modelos basados en las regresiones y las redes neuronales.
- **Lazy**: cada una de las instancias establece una distancia (en algún caso, por ejemplo, distancia euclídea) con el resto de las instancias y escoge a la más cercana. Estos algoritmos están basados en el modelo del "vecino más cercano".
- **Meta**: en este grupo se combinan distintos tipos de aprendizaje con el fin de mejorar el porcentaje de acierto. Se toman distintos parámetros, para hacer convertir a estos clasificadores en poderosos aprendizajes.
- **Trees**: quizás uno de los grupos más importantes de este apartado. Se trata de algoritmos basados en la construcción de árboles de decisión. Estos árboles se construyen mediante los valores que van tomando los atributos y cada hoja se divide en dos hojas más para así acotar el resultado. Algunos de los más destacados son Random Forest o J48.

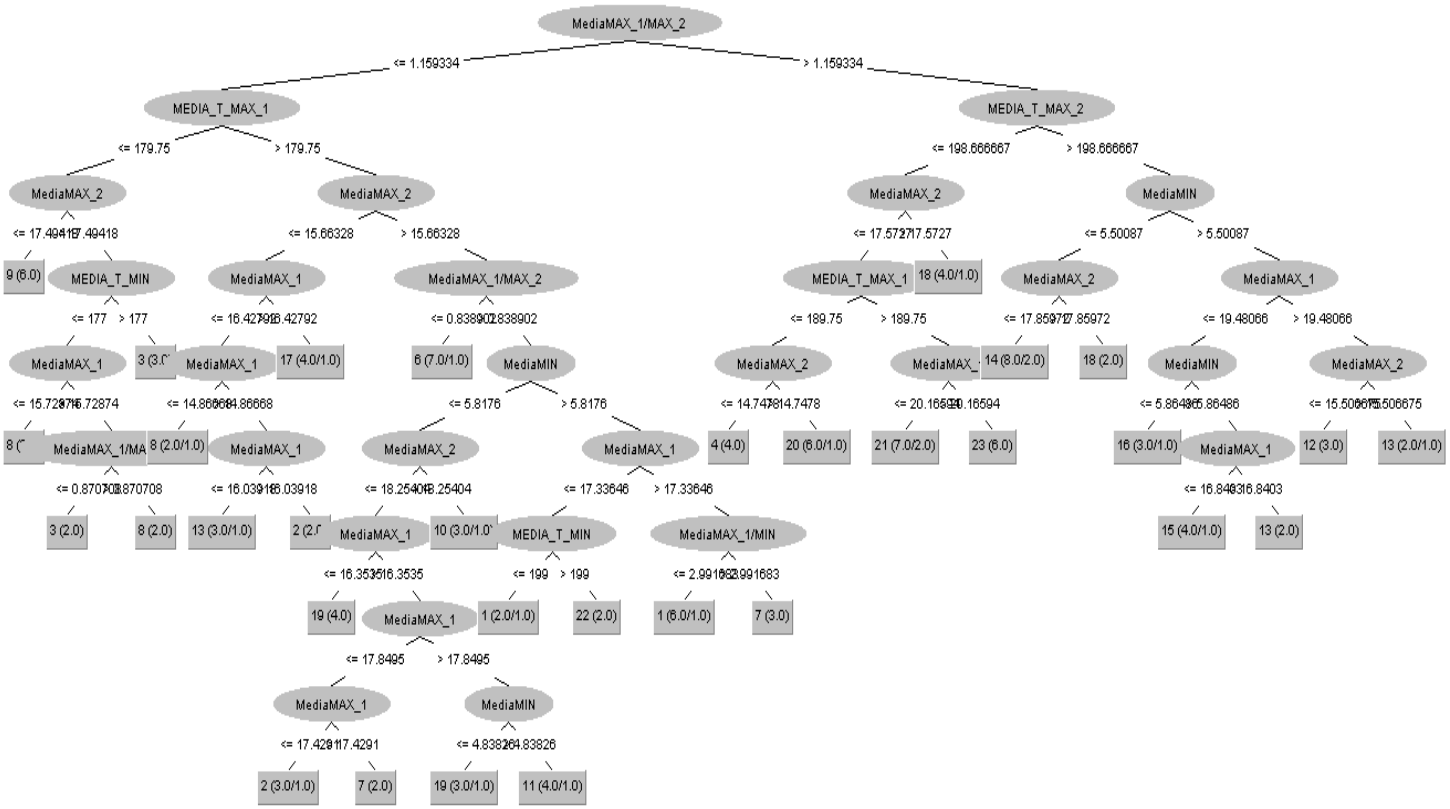

Figura 11. Ejemplo árbol de decisión.

- **Rules**: algoritmos basados en reglas en las que predominan el aprendizaje automático como, por ejemplo, la lógica borrosa. Hay multitud de reglas que combinan todo tipo de estrategias, desde tablas de decisión hasta modelos de regresión a partir de árboles de decisión.

La segunda ventana que hay que destacar se corresponde con las distintas opciones que pueden ser usadas para el tratamiento de los datos.

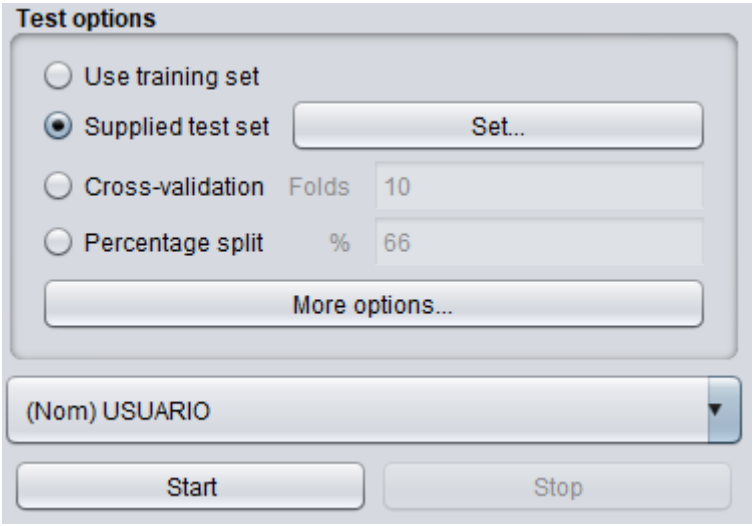

Figura 12. Ventana *test options*.

En el menú de la imagen superior se podrán elegir entre 4 opciones bien diferenciadas: [12]

- **Use training set**: esta opción sirve para que el programa utilice todos los datos de entrada como un solo conjunto y después realice una evaluación sobre este mismo conjunto de datos.
- **Supplied test set**: para esta opción es necesario tener los datos divididos en dos ficheros completamente distintos. En primer lugar, se realizará un entrenamiento sobre los datos introducidos al principio del programa y después se realizará una comparación o evaluación sobre el fichero elegido y cargado en la opción *set*.
- **Cross-validation**: en esta opción el programa realiza una evaluación de los datos conocida como validación cruzada que utiliza un análisis estadístico para garantizar que los datos de entrenamiento y de prueba (test) son independientes. Esta técnica repite y calcula la media de cada evaluación sobre diferentes carpetas o particiones, que se podrán seleccionar en la opción de *folds*.
- **Percentage Split**: con esta última opción, WEKA divide los datos según el porcentaje especificado por el usuario en la opción desplegable *%* para así establecer una cantidad de los datos para el aprendizaje y otra para la evaluación.

Por otro lado, la tercera ventana consistirá en una lista histórica de todas las pruebas hechas con distintos clasificadores llamada *result list*.

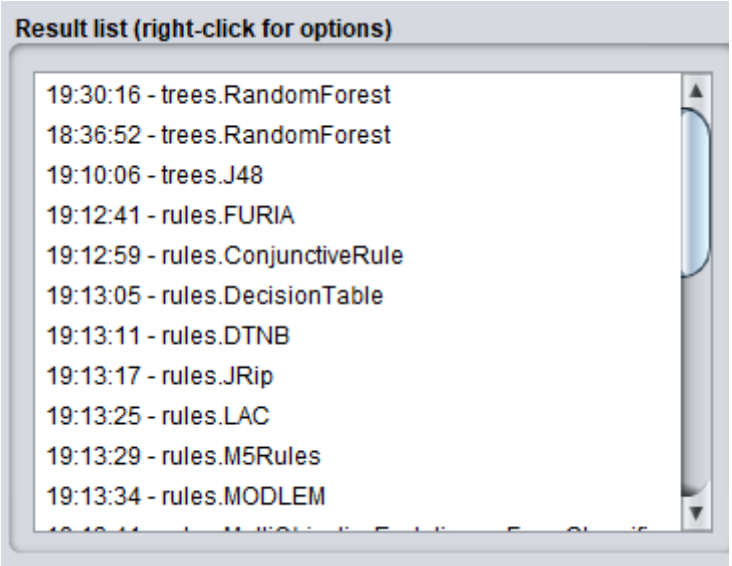

Figura 13. Ventana result list.

Hay que destacar ciertas opciones que existen sobre este menú, en el que se pueden encontrar alguna como visualizar la curva del umbral, representación del árbol, curva o análisis de cote/beneficios o incluso visualizar en los errores de clasificación.

Esta última, es muy interesante, ya que se pueden ver mediante una gráfica simple de dos ejes las instancias incorrectamente clasificadas, así como los parámetros y la confusión por la que han sido malinterpretadas por WEKA.

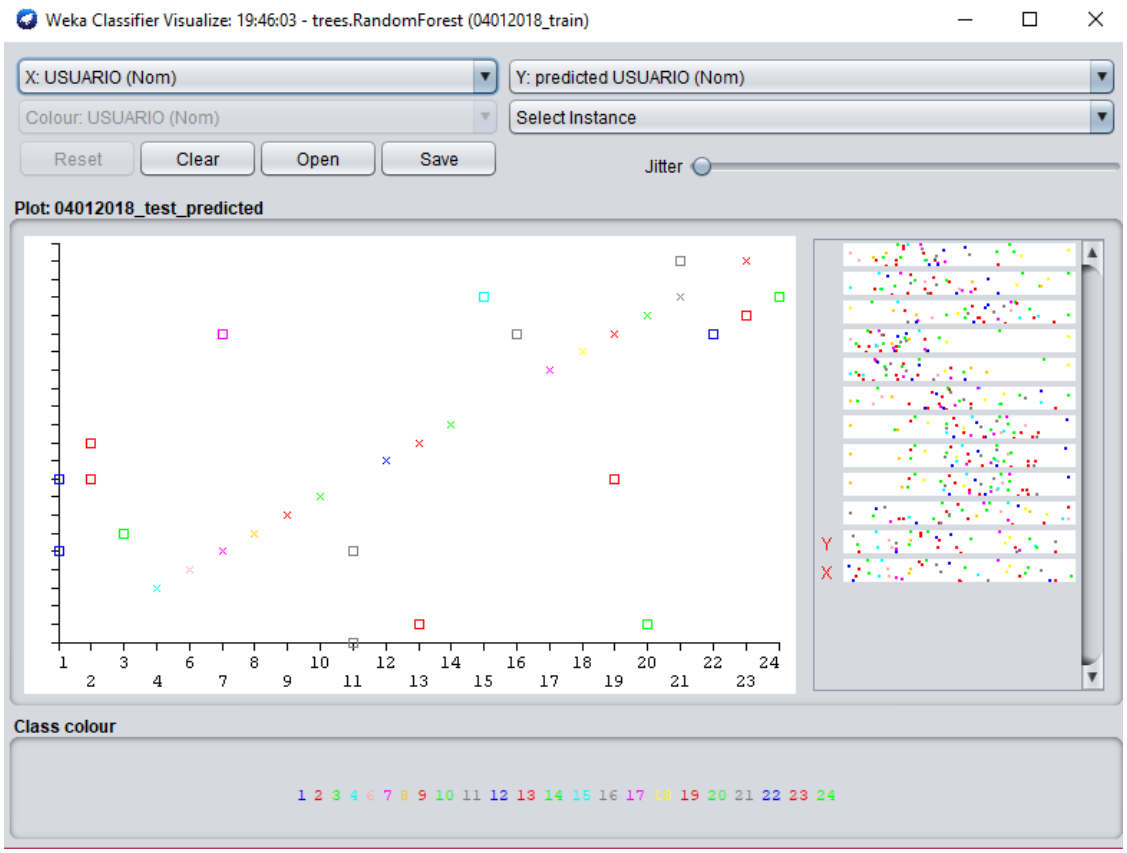

Figura 14. Visualización de los errores de clasificación.

Analizando la imagen anterior, se puede observar una gráfica que enfrenta, la muestra realmente de la instancia con respecto a la muestra predicha por el clasificador. Las muestras correctamente clasificadas se representan en una diagonal lineal con una 'x' y las incorrectas con un □.

Esto puede llegar a ser muy útil, ya que, pinchando sobre las instancias incorrectamente clasificadas, el programa te ofrece todos los atributos de esa instancia para que puedas compararlos con las demás.

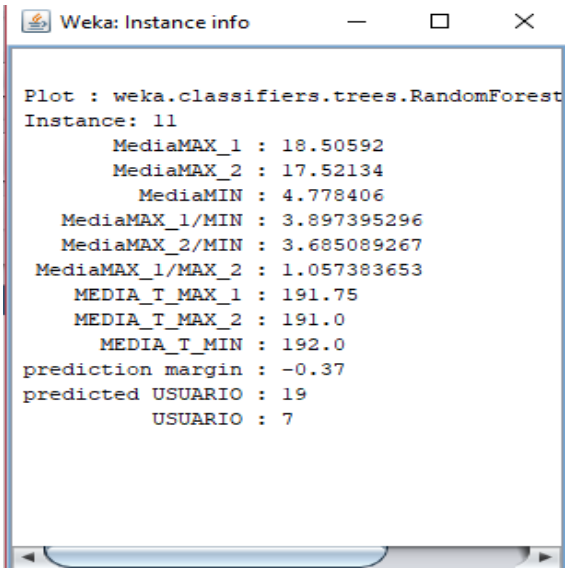

Figura 15. Información de los atributos de cada instancia.

Por último, la ventana quizás más importante de la pestaña de *classify*, llamada *classifier output*. Se trata de la salida del programa y en ella se podrán observar todos los resultados, análisis y evaluaciones llevadas a cabo por el clasificador escogido.

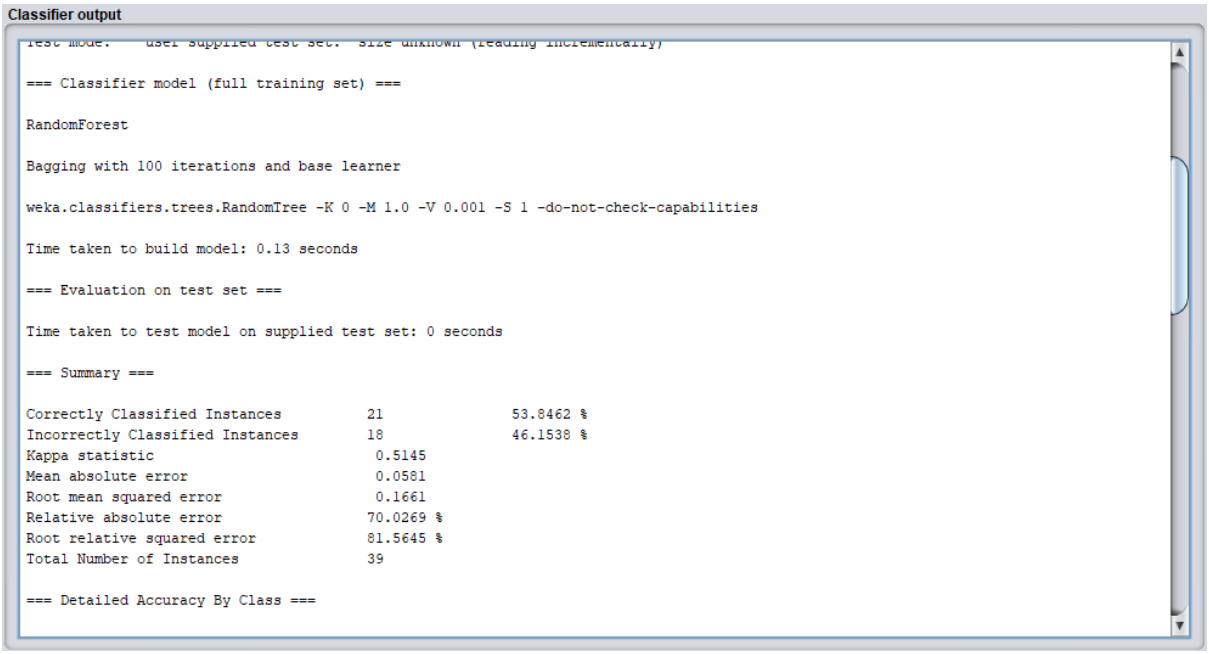

Figura 16. Ventana *classifier output*.

Si se desglosa el *output* mencionado, se pueden apreciar distintas partes o epígrafes que ayudan a comprender y comunicar el resultado de la evaluación.

Dichos apartados son las siguientes:

- **Run information**: aquí se resume las características del conjunto de datos (nombre, numero de atributos y numero de instancias) y la opción de prueba elegida por el usuario para analizar los datos.
- **Classifier model**: en este apartado se puede observar todo lo relacionado con el clasificador elegido, es decir, nombre, numero de iteraciones y tiempo necesitado para construir el modelo.
- **Evaluation on test set**: muestra el tiempo necesitado para comparar con la base de datos usada como *supplied test set* en caso de que la hubiera.
- **Summary**: en este apartado se resume los porcentajes de acierto y error de la evaluación. Se expresan tanto en número de instancias correctamente clasificadas como en errores absolutos y relativos.
- **Confusion matrix**: muestra una matriz cuadrada en la que el número de filas y columnas se corresponde con el número de distintos atributos existentes en cada muestra que se quieren clasificar. Si la muestra esta correctamente clasificada se sumará en la diagonal de la matriz y si no lo estuviera se sumaría en el atributo por el que se ha confundido. Esta matriz da una información muy útil porque refleja los errores producidos y además también informa del lugar donde se han cometido.

4 Diseño de la solución

En este apartado va a ser analizado el problema de reconocimiento de usuarios a través de la pisada gracias al aprendizaje automático de algoritmos de machine learning.

Una vez analizado, se propondrá la solución escogida detalladamente, para así, describir el diseño de dicha solución anteriormente propuesta.

Para explicar el diseño de esta solución, se propondrán distintos puntos en los cuales se expondrá la manera en la que han sido ideados y un análisis en profundidad de los aspectos más relevantes de cada uno de ellos.

4.1 Planteamiento del problema

La finalidad de este trabajo es, como se ha explicado anteriormente, el reconocimiento de usuarios a través del análisis de la marcha o también conocido como gait.

El análisis de la marcha aún está siendo estudiado en la actualidad y no está desarrollado en profundidad, lo que hace de este trabajo algo novedoso y a su vez más complicado a la hora de obtener resultados claros.

Para llevar a cabo este reconocimiento el trabajo parte de una base de datos dada a partir de la cuál, se deberá conseguir un porcentaje de acierto de reconocimiento suficiente como para distinguir dos usuarios diferentes.

Para ello se necesitará realizar un estudio de la base de datos dada y analizarla gráficamente, donde será necesario observar las características que hacen de la pisada algo único en cada usuario.

Una vez obtenidas las características propias de cada usuario, estos datos deberán ser exportados a un programa para recogerlos y agruparlos en una nueva base de datos con el fin de obtener diferentes medidas estadísticas.

Esta nueva base de datos se introducirá en un programa de minería de datos, en este caso será WEKA y será el encargado de realizar el análisis y la evaluación de estos datos para así estimar la fiabilidad de este análisis.

4.2 Diseño de la solución

Antes de entrar en detalle acerca del desarrollo del trabajo, es necesario establecer un diseño al problema planteado en torno al cual gira el núcleo de este trabajo.

A continuación, en este apartado se expondrán una serie de ventajas y desventajas de cada una de las partes del diseño del trabajo. De esta manera será más sencillo llegar a aquella solución que pueda resultar más eficiente en cuanto a los objetivos perseguidos.

Para facilitar la comprensión de este punto, se ha desarrollado un esquema en el que se enuncian las diferentes partes de este punto con sus correspondientes explicaciones en detalle.

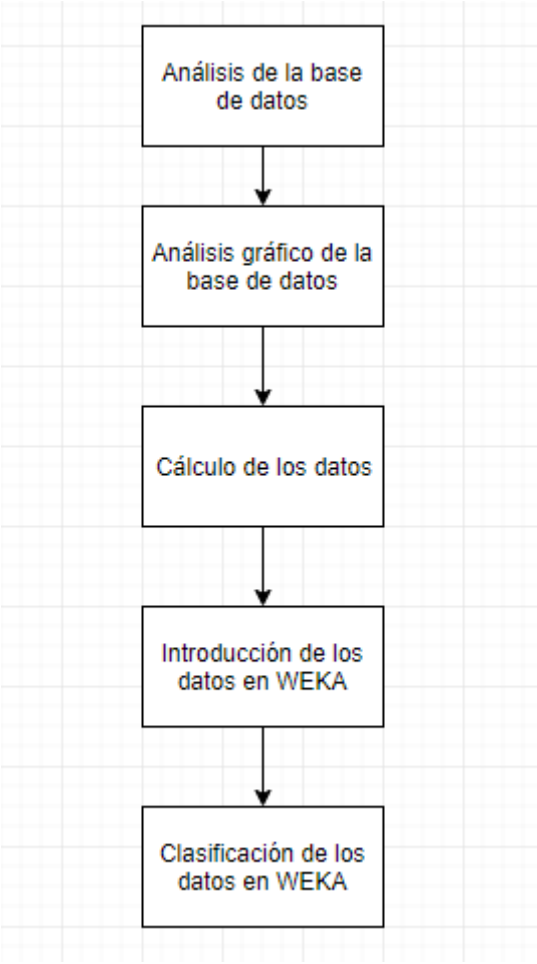

Figura 17. Diagrama del diseño de la solución.

4.2.1 Análisis base de datos

En primer lugar, hay que detallar que en este trabajo no ha sido necesario una recogida de muestras de usuarios en una base de datos. La base de datos de la que se dispone ha sido recogida previamente en otro estudio y el objetivo será entonces, simplemente analizarla y trabajar con ella.

Esta base de datos se compone de 23 archivos distintos correspondientes a 23 usuarios distintos. El nombre de dichas carpetas será de 7000**01**, 7000**02**, … ,7000**23** y dependiendo de cada una de ellas tendrán dentro un numero distinto de ficheros de texto que podrá oscilar entre 3 y 8.

Cada fichero de texto supone una medición de datos distinta tomada al usuario correspondiente durante un tiempo determinado y bajo unas circunstancias que cambiarán en cada fichero.

Dichas circunstancias se recogen y especifican en el nombre de cada fichero la siguiente nomenclatura: tr70_AC_700013_S3_C1-01_999_0_SWS6N.

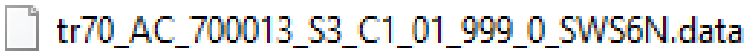

Figura 18. Ejemplo nomenclatura de los ficheros de la base de datos.

Es importante conocer en profundidad el significado de cada sigla para poder entender las posibles diferencias entre las medidas de un mismo usuario.

En las siguientes tablas explican cada parte del nombre de los ficheros:

Tabla 2. Significado de la nomenclatura de los ficheros.

Device		Sensor	_	UserId	_	Process	_	Visitid	_	Quality	_	Error	_	Activitytype	Clothes	Footwear	Deviceposition	Visitconditions
DDDD	_	CC	_	UUUUUU	_	SS	_	MM	_	QQQ	_	Z	_	T	T	T	T	T
(example)																		
BQ01	_	AC	_	000001	_	VV	_	01	_	999	_	0	_	U	P	B	F	T

sensor	process	activityType	clothes		smartphone position	visit conditions	sports activity level (no existe en el nombre del archivo, sólo en la base de datos)	attire (no existe en el nombre del archivo, sólo en la base de datos)
			clothes	footwear				
AC - Accelerometer GR - Gravitational GY - Calibrated gyroscope GU - Uncalibrated gyroscope VR - Relative velocity MF - Magnetic field OR - Orientation SC - Step counter SD - Step detector AU- Grabación de Audio Vi- Grabación de video	VI = Visita S0 = Escenario 0 (Libre) S1 = Escenario 1 S2 = Escenario 2	T - Formación W - Andando R - Corriendo U - Subiendo Escaleras D - Bajando Escaleras M - Variado S - Escenario	P - Vaqueros W - Chándal K - Falda S - Pantalón corto U - Uniforme O - Otro	B - Botas P - Deportivas T- Deportivas S - Zapatos H - Tacones altos E - Tacones bajos F - Chanclas O - Otro	1 - Brazo derecho 2 - Brazo izquierdo 3 - Tobillo derecho 4 - Tobillo izquierdo 5 - Cinturón derecha 6 - Cinturón izquierda 7 - Cinturón atrás 8 - Bolsillo delantero derecho 9 - Bolsillo delantero izquierdo A - Bolsillo trasero derecho B - Bolsillo trasero izquierdo C - Bolsillo de la camisa O - Otro	T - Lesión temporal P - Embarazo D - Ebriedad O - Otro N-none	N - Nada M - Una vez al mes o menos W - Una vez a la semana T - Dos veces a la semana o más D - Diariamente	case "1": clothes = "U - Uniforme"; footwear = "B - Botas"; case "2": clothes = "W - Chandal"; footwear = "S - Shoes"; case "3": clothes = "P - Vaqueros"; footwear = "S - Shoes"; case "4": clothes = "P - Vaqueros"; footwear = "T - Zapatil case "5": clothes = "O - Otro"; footwear = "O - Otro";

Viendo todos los ficheros de los distintos usuarios, se puede apreciar que, en el caso de esta base de datos, todas las aceleraciones han sido tomadas por acelerómetros.

En cuanto al contenido de cada archivo, todos ellos siguen el mismo formato:

```
1457684947071;9.5803990;-2.1021090;-0.7134720
1457684947072;9.2978820;-0.5123590;-1.3263880
1457684947072;9.2978820;-0.5123590;-1.3263880
```

Figura 19. Ejemplo del formato de los datos del fichero.

En primer lugar, se recoge el tiempo en el que ha sido tomado la muestra y, separados por puntos y comas, aparecen las aceleraciones tomadas por el acelerómetro en 'x' 'y' y 'z' sucesivamente.

4.2.2 Análisis gráfico base de datos

Para poder analizar la base de datos dada, se ha estimado que la opción más ventajosa es representar gráficamente cada muestra de cada usuario (ya que queda descartado analizar los datos sin poder observarlos) y así poder examinar sus características comunes y excepcionales de los usuarios.

Se decide ver los datos representados gráficamente por la facilidad que se tiene al poder observar las gráficas de todos los usuarios y saber cuál es el patrón que se va a repetir en todos los usuarios.

Para llevar a cabo esta tarea y poder mostrar las gráficas, es necesario agrupar las aceleraciones para poder representarlas frente al tiempo.

Debido a esto, este punto se va a dividir en dos subpuntos bien diferenciados. Por un lado, la manera en la que se agruparán estas aceleraciones y su cálculo mediante un programa en Java y por otro lado la representación de éstas y su estudio.

4.2.2.1 Agrupación de los datos

Lo primero que se necesita es saber cuál es la manera en la que se van a agrupar los datos para su posterior estudio y representación. Se ha demostrado que el cálculo de la magnitud es ventajoso para las características de la aceleración en estudios de identificación de la marcha basados en dispositivos móviles y a menudo se han utilizado en investigaciones existentes para manejar las variaciones en la colocación del sensor.

Se han contemplado también otras opciones como la media aritmética, pero desde el principio se ha utilizado el cálculo de la magnitud debido a estudios previos en los que se demuestra el poder de la magnitud en esta clase de problemas.

Una vez se ha decidido la magnitud como manera de agrupar los datos, es necesario la creación de un programa que lea todas las líneas de todos los ficheros de los usuarios y nos dé como resultado una base de datos basada únicamente en dicha magnitud.

En un principio se pensó la posibilidad de trabajar en lenguaje C++, ya que ha sido el estudiado a lo largo del grado con mayor frecuencia, pero finalmente se ha decidido la creación de un programa en lenguaje Java debido a su facilidad de uso y la incorporación de librerías y métodos que ayudarán a diferenciar los datos separados por punto y coma.

El programa que se usará para calcular la magnitud de la base de datos dada funciona según el siguiente diagrama:

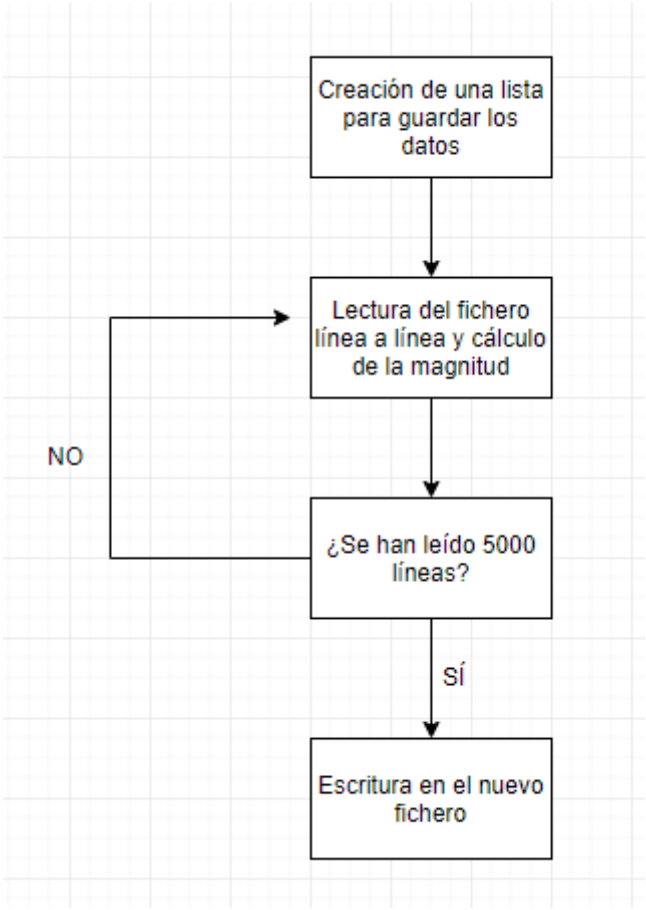

Figura 20. Diagrama del diseño del programa para la agrupación de los datos.

En este programa creado, es importante identificar el número de medidas por cada muestra de cada usuario que vamos a calcular, ya que la base de datos contiene demasiados datos para representar.

En un principio se pensó la opción de calcular las 1000 primeras muestras, pero su principal inconveniente es que tanto si se cogen las 1000 primeras como las 1000 últimas, es muy posible que los datos sean muy irregulares, ya que éstos se corresponden al inicio y fin de la marcha del usuario. Por tanto, se ha decidido coger las primeras 5000 muestras y calcular la magnitud de todas ellas.

4.2.2.2 Representación y análisis de las gráficas

Una vez se tiene esta nueva base de datos compuesta por la magnitud de todas las muestras de los usurarios es necesaria su representación por lo que acudiremos a Excel para ello.

Se ha elegido Excel para la representación de las gráficas debido a su libre disposición en cualquier ordenador y su facilidad de uso.

Antes de proceder a la representación de las gráficas, es de vital importancia saber cuántas muestras se van a representar. La primera opción pensada fue representar las 5000 muestras calculadas según el punto anterior, pero la gráfica quedaba muy compacta y las primeras medidas eran irregulares.

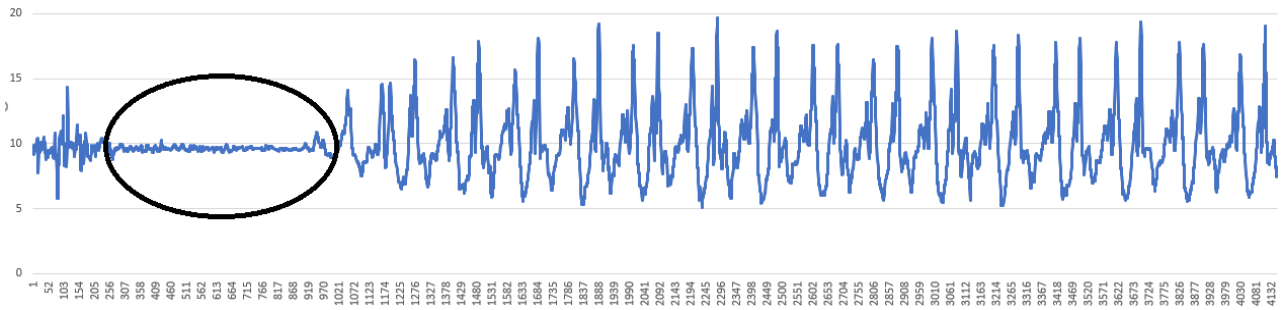

Figura 21. Representación gráfica de los datos (5000 muestras).

El siguiente paso, por tanto, fue reducir el número de muestras a representar, por lo que se optó por coger las últimas 1000 muestras de conjunto de las 5000 calculadas. De esta manera se evitaba tener medidas irregulares y que la gráfica dejara de ser compacta, para así poder observar bien sus características propias.

Una vez representadas las gráficas, es necesario distinguir ciertos parámetros de las gráficas que serán de importancia en el desarrollo posterior del trabajo. Estos parámetros ayudarán a diferenciar unas pruebas de otras, ya que las diferentes muestras que posee un mismo usuario deben ser parecidas y tener valores cercanos.

Se puede observar a lo largo de las representaciones de todos los usuarios un patrón común en todas ellas en las que algunos de sus parámetros principales se pueden apreciar en la siguiente imagen:

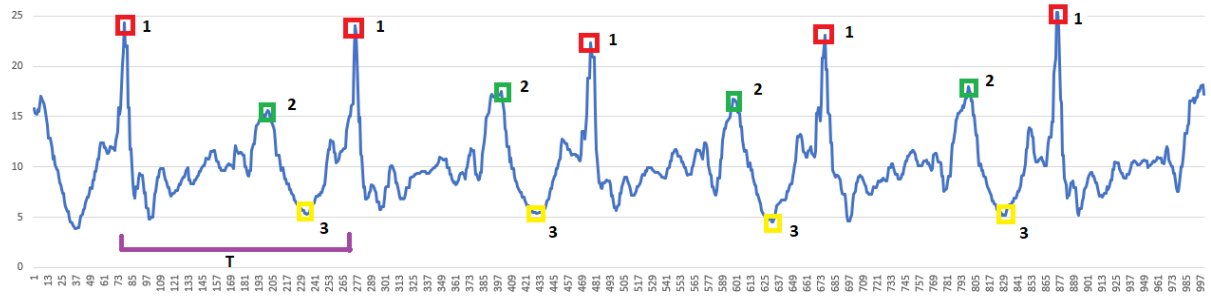

Figura 22. Representación gráfica y parámetros de los datos (1000 muestras).

Todas las funciones representadas poseen dos máximos bien diferenciados que destacan por encima del resto. En el caso de este ejemplo, el máximo número 1 (color rojo) es más alto que el máximo número 2 (color verde), aunque esto no es una regla que se ha de cumplir siempre, ya que habrá veces que el máximo 2 sea superior al máximo 1 o incluso sean iguales.

Por otro lado, de color amarillo, hay que resaltar el mínimo absoluto que poseerán todas las representaciones graficas de la marcha. Siempre irá situado inmediatamente después del máximo número 2.

Por último, también es importante tener en cuenta el periodo de la muestra, ya que variará en cada usuario y será útil para su análisis.

4.2.3 Cálculo de los parámetros de los datos

En primer lugar, se deben elegir los parámetros característicos de las representaciones graficas de los datos. En este caso se basarán en los máximos, mínimos y periodo, como ha sido explicado en el punto anterior.

Simplemente con estos 4 datos por gráfica no es suficiente para distinguir a todos los usuarios, ya que muchos de ellos comparten ciertas medidas de algunas de las características. Por eso se realizarán diferentes operaciones matemáticas con el fin de aportar un abanico de datos y medidas mayor.

Los parámetros que se van a obtener de cada muestra de los usuarios recogidos en la base de datos son las siguientes:

- La media aritmética de todos los máximos 1.
- La media aritmética de todos los máximos 2.
- La media aritmética de todos los mínimos.
- La media aritmética del periodo entre los máximos 1.
- La media aritmética del periodo entre los máximos 2.
- La media aritmética del periodo entre los mínimos.
- El cociente entre la media aritmética de los máximos 1 y los máximos 2.
- El cociente entre la media aritmética de los máximos 1 y los mínimos.
- El cociente entre la media aritmética de los máximos 2 y los mínimos.

Obteniendo la media aritmética de los distintos parámetros se pretende aglutinar todos los parámetros referidos a la misma característica en un solo atributo. Así se pretende reducir el número de atributos por muestra y obtener una mayor simplicidad y garantía de reconocimiento por parte del clasificador.

4.2.4 Introducción datos a WEKA

Antes de introducir la tabla de Excel creada con todos los parámetros de los usuarios calculados es necesario explicar cómo hay que estructurar el fichero con dichos parámetros para que sea leído correctamente por WEKA.

WEKA ofrece principalmente dos opciones a la hora de leer un fichero proveniente de programa de cálculo como Excel:

- Importar un archivo con formato .csv.
- Importar un archivo con formato. arff.

El primero de ellos, únicamente necesita que el usuario elija la extensión al guardar los datos en Excel, mientras que la segunda opción, es un formato propio de creación de bases de datos para WEKA.

En el caso de este TFG, se ha elegido la segunda opción, ya que como va a estar muy centrado en WEKA, es importante conocer todas las partes posibles.

En primer lugar, para introducir el fichero en WEKA, es necesario conocer el formato que debe tener para que el programa lo reconozca y pueda trabajar con él.

Para empezar, todo archivo .arff debe estar encabezado con la sentencia *@relation* acompañada de un nombre. Esto será el nombre de la base de datos y se hará referencia a ella en el programa.

A continuación, se suceden los atributos del fichero, es decir, todos los parámetros calculados en el punto anterior. Todos ellos, en separadas líneas, deberán ir precedidos por la sentencia *@attribute* "nombre del parámetro o atributo" y finalizando con el tipo de dato. En este caso, todos los datos que introduciremos serán de tipo numeric.

Al final de todos los atributos es necesario añadir uno más que se corresponderá con el usuario que se quiere identificar por medio de WEKA.

Para terminar, el último apartado de este archivo se corresponde con los datos. Cada muestra deberá ir en líneas separadas y todas ellas precedidas una sola vez por la palabra *@data*. Los números de cada línea de datos deberán ir separados por comas y el orden en el que estén escritos se corresponderá con el mismo orden en el que aparecen los atributos en el punto anterior.

```
@relation BDWEKA

@attribute MediaMAX_1 numeric
@attribute MediaMAX_2 numeric
@attribute MediaMIN numeric
@attribute MediaMAX_1/MIN numeric
@attribute MediaMAX_2/MIN numeric
@attribute MediaMAX_1/MAX_2 numeric
@attribute MEDIA_T_MAX_1 numeric
@attribute MEDIA_T_MAX_2 numeric
@attribute MEDIA_T_MIN numeric

@attribute USUARIO {'1','2','3','4','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24'}


@data

17.71082,16.10068,6.263292,2.841329609,2.589245294,1.10079745,208.5,207.25,212.25,1
17.47738,17.15588,6.101512,2.864293463,2.813163965,1.019738592,197.5,197.75,199,1
18.8476,16.66188,6.35929,2.975986842,2.63015324,1.133378025,194.75,195.5,196.5,1
16.7905,17.5727,5.846582,2.871866298,3.01068989,0.957065195,194.75,194.75,196,1
17.8495,17.74292,6.197064,2.88633385,2.866672648,1.006200572,191.5,190.25,191.5,1
17.36136,17.20854,6.210206,2.806861721,2.780145684,1.010450553,198.75,199,199.25,1
17.4291,16.56332,5.308446,3.281778252,3.123438661,1.052706417,188,188.75,188.25,2
16.42764,14.25778,6.4304,2.556076661,2.218450213,1.154489147,204,204.75,202.25,2
15.33826,14.89226,6.13833,2.502724426,2.428696912,1.031622498,192.75,191.5,190.5,2
16.8403,17.34038,5.95103,2.858519453,2.993696168,0.97321266,202.25,204.75,198.3333333,2
16.16786,15.4387,5.78196,2.805001727,2.636225059,1.045112336,199.75,200,201,2
16.41138,17.66074,5.147368,3.194150271,3.439023475,0.929217002,199.5,199.5,199.75,2
15.8807,18.29044,4.767448,3.33517926,3.837700137,0.870708478,170.5,214.25,170.75,3
16.67018,18.75238,5.38112,3.135354439,3.510535406,0.890524248,178,178,177.75,3
```

Figura 23. Formato del fichero .arff.

Una vez se ha completado el archivo .arff que será introducido en WEKA, se debe decidir cómo se quiere realizar el inicio del aprendizaje automático para el posterior reconocimiento de los datos.

Hay que destacar en primer lugar si se va a optar por una única base de datos o por el contrario de dos bases de datos, siendo una de entrenamiento y otra de prueba (test).

En el caso de este trabajo, se ha optado por separar los datos en dos bases de datos diferenciadas y así poder usar de ellos para su entrenamiento y conseguir con el resto un porcentaje mayor de acierto durante la prueba. Se ha establecido un porcentaje de entrenamiento-prueba de 70-30% de los datos, ya que es lo habitual en estos casos.

4.2.5 Clasificación de los datos en WEKA

Una vez terminados e introducidos los ficheros en WEKA, el siguiente paso será clasificar todas las instancias de éstos para proceder a ejecutar la evaluación y examinar los resultados del reconocimiento.

Para llevar a cabo esta clasificación, lo primero que se debe decidir es como va a ser el tratamiento de los datos. En este trabajo, tal y como se ha especificado en el apartado anterior, existirán dos bases de datos, una para entrenar y otra para testear.

Una vez introducidas las bases de datos y quizás la parte más importante de este punto, es elegir los clasificadores que van a ayudar al desarrollo de este punto. Sin embargo, para decidir cuál de todos estos clasificadores elegir, se debe analizar qué tipo de datos tenemos para así, escoger uno u otro.

A la hora de elegir dichos clasificadores, se ha optado por usar dos de los más usados en este tipo de trabajos como son NaiveBayes y RandomForest, ya que están basados en algoritmos de machine learning bastante extendidos y se han usado en infinidad de veces en otros sistemas de clasificación.

A continuación, se va a explicar el porqué de esta elección de éstos entre tantos otros clasificadores posibles:

- **RandomForest**: es quizás uno de los árboles de decisión más utilizados en el ámbito del ML. Se ha escogido dicho árbol por su robustez frente a otro tipo de clasificadores ante el ruido.
- **NaivesBayes**: los datos están formados por un conjunto de atributos numéricos y cuando se quiere clasificar en un conjunto finito de clases, se acude a un clasificador de tipo bayesiano. Este algoritmo asume por su parte, que el valor de cada atributo es independiente del valor del resto de atributos. Es por esto por lo que, en este caso, se simplifican los cálculos consiguiendo un porcentaje de acierto mayor en la clasificación realizada.

5 Desarrollo de la solución

En el presente apartado se va a explicar paso a paso todo el desarrollo de este trabajo, exponiendo detalladamente desde el cálculo de los ficheros hasta el resultado final de la clasificación. A lo largo de este punto también se van a describir los métodos empleados durante toda la realización de dicho trabajo y las soluciones derivadas del diseño propuesto anteriormente.

Para hacer su lectura más sencilla, al igual que los anteriores apartados, se ha dividido éste en distintos epígrafes que facilitarán la comprensión de todo lo expuesto.

5.1 Adaptación y cálculos sobre la base de datos original

Como se ha explicado anteriormente, este trabajo parte de una base de datos ya recogida previamente de forma independiente a la realización de éste.

Esta base de datos contiene los ficheros donde se recogen todas las muestras de las pruebas y mediciones realizadas a los usuarios. Es necesario adaptar estas muestras para este trabajo, ya que como se ha explicado en el punto del diseño de la solución, no se va a trabajar con aceleraciones y tiempos, sino con magnitudes.

Lo primero que habrá que hacer entonces, será convertir dichas aceleraciones procedentes de los acelerómetros del dispositivo con el que se han realizado las mediciones en la raíz de la suma de sus cuadrados, es decir, su magnitud. Para ello se ha realizado un programa en Java que ayudará a dicha conversión.

En lo referente al programa mencionado anteriormente, lo principal que hay que destacar son las clases de Java que se han incluido para su correcto funcionamiento.

Las clases añadidas son las siguientes:

```
3  import java.io.BufferedReader;
4  import java.io.BufferedWriter;
5  import java.io.File;
6  import java.io.FileNotFoundException;
7  import java.io.FileReader;
8  import java.io.FileWriter;
9  import java.io.IOException;
10 import java.nio.file.Path;
11 import java.nio.file.Paths;
12 import java.util.ArrayList;
13 import java.util.StringTokenizer;
14 import java.util.logging.Level;
15 import java.util.logging.Logger;
```

Figura 24. Clases importadas para el programa en Java.

De entre todas estas clases se pueden destacar algunas de ellas como el *StringTokenizer* que servirá para separar una cadena en partes (*tokens*) o *Arraylist*, que permitirá crear una lista para almacenar los datos.

Las demás clases importadas son utilizadas universalmente en la mayoría de los programas desarrollados en Java para funciones básicas como es el caso de los *Buffer* para escribir y leer en ficheros o *IOExceptions* para crear excepciones.

En cuanto al desarrollo del programa lo primero que se puede observar dentro del *main()* del programa es la creación de dos objetos; uno es tipo *path* que nos ayudara a mapear la ruta donde se encuentran nuestros ficheros y el otro es tipo *file* que transforma el objeto tipo file para poder tratar con él.

```
35      Path path = Paths.get("D:\\Data\\");
36      File file = path.toFile();
37      File[] folders = file.listFiles();
```

Figura 25. Creación de objetos.

Después de esto es necesaria la creación de un *arraylist*, es decir, un vector o lista donde se guardarán posteriormente los datos. Según se vayan guardados datos en este vector, su memoria se irá incrementando.

```
38      ArrayList<String> lineasList = new ArrayList<String>();
```

Figura 26. Creación de la lista.

A continuación, se creará un objeto que irá leyendo línea por línea los ficheros llamados *bf* y llamará a la función *processlinea()*. Esta función recibe la línea leída anteriormente y gracias al método *tokenizer* la separa por comas, para así después calcular la magnitud y devolver el resultado.

El cálculo de la magnitud se obtendrá gracias a la función matemática que posee Java llamada *Math.sqrt()*.

```
97      private static String processLinea(String linea) {
98          String result = "";
99          StringTokenizer str = new StringTokenizer(linea, ";");
100         if (str.countTokens() == 4) {
101             str.nextToken();
102             double a = Double.parseDouble(str.nextToken());
103             double b = Double.parseDouble(str.nextToken());
104             double c = Double.parseDouble(str.nextToken());
105             double d = Math.sqrt((a*a)+(b*b)+(c*c));
106             result = Double.toString(d);
107         }
108         return result;
109     }
```

Figura 27. Función Math.sgrt() para el cálculo de las magnitudes.

Para tener la garantía de que todo esto se repetirá las veces que se indique, dentro del bucle *while* hay una condición *if-else* en la que se llamará a la función *processlinea()* tantas veces como sea necesario. En el caso de este programa se ha optado por leer 5000 líneas, por lo que tendremos 5000 magnitudes.

```
while (linea != null) {
    String lineaFromFile = processLinea(linea);
    //Hacerlo para los 1000 primeros, volcar al mismo fichero
    if (i < 5000) {
        lineaToFile = lineaToFile.concat(lineaFromFile).trim().concat(",");
    } else if (i == 5000) {
        lineaToFile = lineaToFile.concat(lineaFromFile).trim().concat(" --- ").concat(folder.getName());
    } else {
        break;
    }
    i++;
```

Figura 28. Bucle while.

Por último, se creará el fichero de salida (output.txt) donde se escribirán todos los nuevos datos. Para llevar a cabo dicha escritura será necesaria la creación de un objeto que nos ayude a ello, que se llamará *bw*.

```
file = new File("D:\\output.txt");
BufferedWriter bw = null;
try {
    bw = new BufferedWriter(new FileWriter(file));
    for (String linea : lineasList) {
        bw.write(linea);
        bw.newLine();
    }
```

Figura 29. Creación del fichero de salida.

Una vez se han calculado todas las magnitudes de tantas líneas como se hayan seleccionado, se obtiene el fichero output.txt, el cuál servirá en el siguiente apartado para calcular todos los parámetros y gráficas de los datos.

5.2 Introducción y representación de los datos en Microsoft Excel

Una vez calculadas todas las magnitudes de los datos, el siguiente paso es la introducción y la representación de estos. Para ello, se ha optado por el uso de Microsoft Excel, ya que ofrece una amplia gama de opciones y su facilidad de obtención y uso.

En primer lugar, antes de calcular los parámetros que posteriormente serán los atributos de los datos, es necesaria la representación gráfica de la magnitud calculada de los datos.

Para ello seleccionaremos en el fichero outuput.txt los 1000 últimos datos como se ha especificado en el punto anterior dedicado al diseño de este trabajo. Una vez seleccionados, se procederá a copiarlos en una hoja en blanco de Excel, pero éstos aparecerán todo juntos, en la misma celda, por lo que es necesario separarlos.

	A	B	C	D	E	F	G	H
1	9.46547,9.45604,9.45371,9.45371,9.4338,9.52958,9.52559,9.55378,9.65486,9.65223,9.66168,9.66274,9.61937,9.61							
2								
3								

Figura 30. Visualización de las magnitudes en Excel.

A continuación, para llevar a cabo dicha separación de los datos en distintas celdas, es necesario acudir a Datos->texto en columnas.

En dicha función de Excel, se seleccionará que los datos estén delimitados con caracteres como comas o tabulaciones, para posteriormente escoger el separador que se tenga en los datos, en nuestro caso la coma.

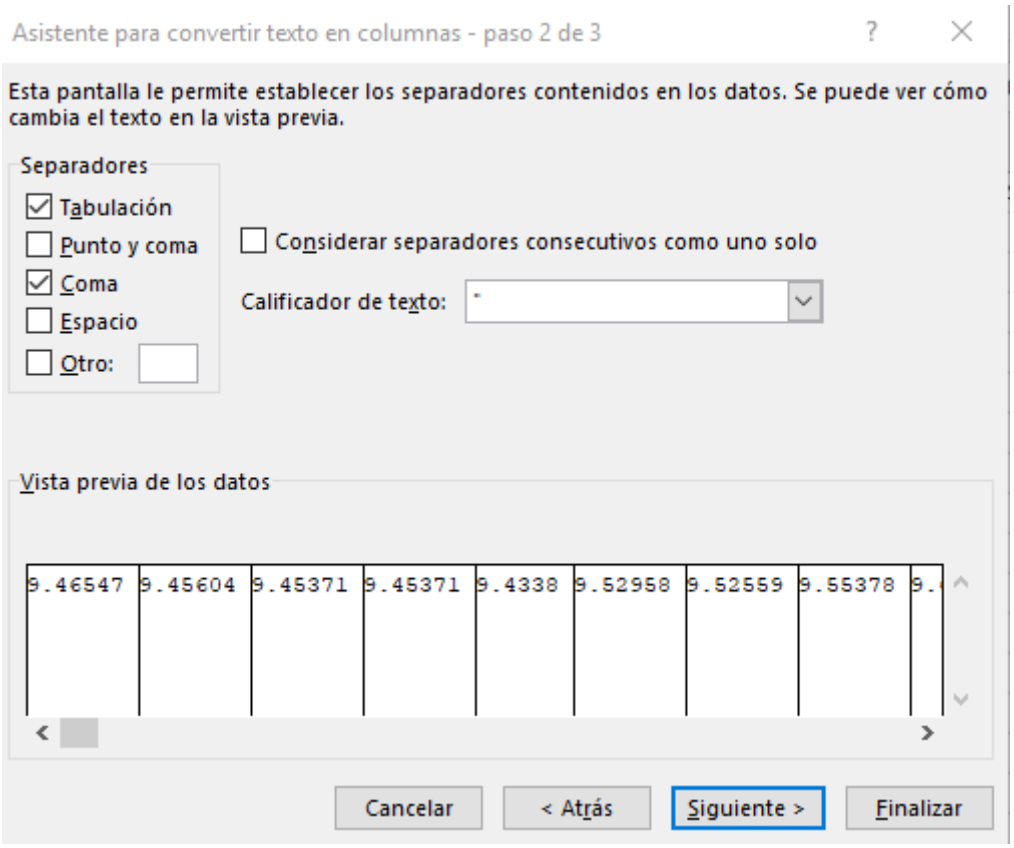

Figura 31. Ventana para separar los datos en columnas.

Una vez finalizado este proceso para todos los usuarios, los datos aparecerán en las 1000 primeras celdas de cada hoja de Excel, de tal manera que el siguiente paso natural será su representación gráfica.

	A	B	C	D	E	F	G	H	I	J	K
1	4,55432	4,54623	6,22617	6,22617	7,37314	7,59544	8,32219	8,03235	8,05868	7,55282	7,601
2											

Figura 32. Representación de los datos en columnas.

Para terminar con este punto, se van a representar las gráficas referentes a la magnitud de todos los usuarios incluidos en la base de datos. Para ello se van a seleccionar todos los datos del

mismo usuario en Excel y pinchando en Insertar->gráfico recomendados->todos los gráficos se elegirá la gráfica que se crea más conveniente. En este caso se ha optado por una gráfica de línea.

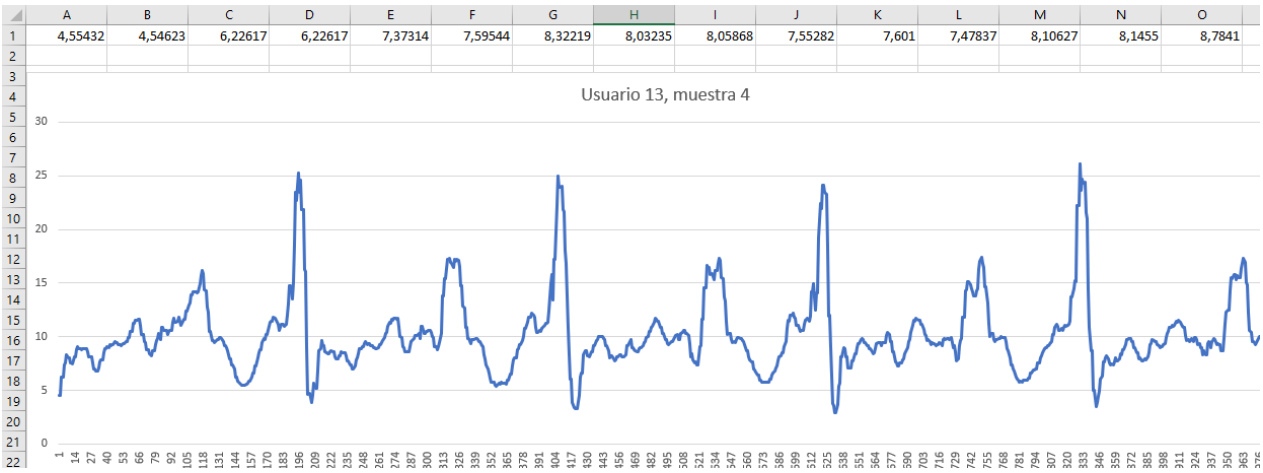

Figura 33. Representación gráfica de los datos.

5.3 Cálculo de los parámetros de los datos

Una vez se tienen todas las gráficas representadas para todas las mediciones de todos los usuarios, el siguiente paso será calcular los parámetros característicos explicados en el punto 4.2.3 de este trabajo.

Para hacer la lectura de este punto un poco más sencilla, se va a dividir en distintos apartados explicando el cálculo de cada uno de ellos por separado.

5.3.1 Cálculo de los puntos máximos

Lo primero que se va a calcular son los puntos máximos ‘1’ y máximos ‘2’ de todas las gráficas. Para ello se va a recurrir a la fórmula que se encuentra en Microsoft Excel y ayudará a dicho cálculo. La fórmula a tratar es la siguiente:

```
=MAX('[GraficasTFG.xlsx]1_8'!$A$1:$AX$1)
```

Figura 34. Fórmula de Excel para calcular los máximos.

Para entender esta fórmula es necesario destacar y explicar las distintas partes de las que se compone:

- **MAX**: hace referencia al nombre de la fórmula que se usa en este apartado. Dentro de los paréntesis se encontrarán los diferentes argumentos en los que se especificarán sus funciones.
- **GraficasTFG.xlsx**: esta parte de la fórmula alude al libro de Excel donde se tienen los datos que se quieren utilizar.
- **1_8**: indica el nombre de la hoja en la que se encuentran los datos a calcular dentro del libro indicado como GraficasTFG. En este caso se corresponde a la muestra 8 del usuario número 1.
- **$A$1:$AX$1**: hace referencia a las celdas de las cuales se quieren calcular el máximo perteneciente al libro y la hoja especificadas anteriormente. En este caso seleccionamos las primeras 100 muestras.

Para no tener que escribir la fórmula anterior cada vez que se quiera calcular un máximo, ya que serían cientos de veces, se ha desarrollado un pequeño programa muy sencillo en C++ que crea un script tantas veces como máximos se quieran calcular.

En lo que respecta al programa, lo primero que es necesario es la creación de un objeto de tipo *ofstream* que cree un archivo de salida en el que escribir los scripts. En este caso, dicho objeto se llamará ScriptMAX y el archivo de salida creado ScriptMAX.txt.

```
ofstream escribirMAX ("ScriptMAX.txt");
```

Figura 35. Creación del objeto para el fichero de salida.

La siguiente parte del programa consta de un doble bucle *for* que irá aumentando de uno en uno dos variables. Dichas variables son referidas a la hoja del libro que se quiere seleccionar, así la variable ‘i’ será para el usuario y la variable ‘j’ para la muestra.

Un ejemplo de lo anterior sería lo siguiente:

- Para i=1, j=1 se obtendrá la hoja de Excel correspondiente al 1_1.
- Para i=1, j=2 se obtendrá la hoja de Excel correspondiente al 1_2
- Para i=1, j=8 se obtendrá la hoja de Excel correspondiente al 1_8
- Para i=2, j=1 se obtendrá la hoja de Excel correspondiente al 2_1.

Y así sucesivamente hasta terminar el usuario número 24.

```
for(int i=0;i<24;i++)
{

    for (int j=0;j<8;j++)
    {

        escribirMAX<<"=MAX('[GraficasTFG.xlsx]"<<(i+1)<<"_"<<(j+1)<<"'!$A$1:$AX$1);=MAX('[GraficasTFG.xlsx]"<<(i+1)
        escribirMAX<<endl;

    }

    escribirMAX<<endl<<endl;
```

Figura 36. Doble bucle *for*.

A continuación, una vez el archivo ScriptMAX.txt está ya rellenado con todos los scripts, lo único que faltaría será seleccionarlos todos y copiarlos en la hoja de Microsoft Excel donde se están calculando todos los máximos. El resultado de esta operación quedaría de la siguiente manera:

	MAXIMOS_1				
	MAX_1	MAX_2	MAX_3	MAX_4	MAX_5
1_1	19,0047	16,9188	16,7123	17,4876	18,4307
1_2	16,8948	17,6557	18,2053	17,2574	17,3737
1_3	20,7385	18,6334	17,849	17,5989	19,4182
1_4	17,3435	15,8006	16,9926	16,5725	17,2433
1_5	16,0203	17,8422	18,8663	18,3094	18,2093
1_6	17,4361	17,2485	17,0711	16,7944	18,2567
1_7	18,5857	17,7862	17,7122	18,1739	17,9446
1_8	20,1233	19,204	19,2226	17,6062	18,3028
2_1	16,7198	18,5931	16,4188	17,346	18,0678
2_2	16,962	16,1921	15,8026	16,9392	16,2423
2_3	14,9809	15,2856	16,1309	15,2021	15,0918
2_4	17,2651	16,5676	15,4509	18,4406	16,4773
2_5	16,3796	16,9739	15,7085	15,3433	16,434
2_6	17,1887	16,2728	15,9343	16,1515	16,5096
2_7	16,8383	18,6389	17,4735	18,3672	16,9215

MAXIMOS_2				
MAX_1	MAX_2	MAX_3	MAX_4	MAX_5
16,8294	16,6855	15,6212	15,1308	16,2365
17,942	17,2969	17,0227	16,9693	16,5485
17,7345	17,0046	17,0877	16,2126	15,27
18,0156	18,4444	17,2896	17,2349	16,879
17,797	17,5666	18,3699	17,3364	17,6447
16,42	17,6314	17,5286	17,7196	16,7431
16,1172	17,1868	16,2956	17,4861	17,5616
18,8894	17,4823	17,8516	18,4536	18,2492
16,7219	16,6051	16,6602	16,6353	16,1941
15,1447	15,0516	13,3083	13,7525	14,0318
14,9561	14,7843	14,7244	15,9195	14,077
18,0931	18,4344	17,0317	17,2799	15,8628
14,7517	15,4058	15,2844	16,3129	
18,1295	17,6589	17,0189	17,686	17,8104
16,2214	16,664	16,4499	17,1953	16,7732
14,9849	13,639	13,3079	13,4867	13,5414

Figura 37. Representación de los máximos.

Por último, como se especificó en el punto del diseño, el parámetro que realmente interesa a la hora de convertirlo en atributo para su posterior introducción en WEKA, es la media aritmética de los máximos.

Para ello se recurrirá a la fórmula de Excel Promedio, la cual hará la media aritmética de las celdas establecidas en el argumento de la función.

Figura 38. Fórmula de Excel para calcular el promedio.

Hay que destacar que, en todo este recorrido de cálculo de los máximos, se obtiene una lista de máximos que se corresponden tanto a máximos '1' como a máximos '2', por lo que es importante empezar todas las gráficas con un máximo '1' y así poder dividir dichos máximos más fácilmente.

5.3.2 Cálculo de los puntos mínimos

En este apartado se van a calcular son los puntos mínimos de todas las gráficas. Este proceso se llevará a cabo de manera muy parecida al cálculo de los máximos del anterior punto. Para ello se va a recurrir a la fórmula que se encuentra en Microsoft Excel y ayudará a dicho cálculo. La fórmula a tratar en este caso será la siguiente:

Figura 39. Fórmula de Excel para calcular los mínimos.

Esta fórmula es prácticamente igual a la que se usa en el caso anterior. Aun así, se destacarán las distintas partes de las cuales se compone:

- **MIN**: hace referencia al nombre de la fórmula que se usa en este apartado. Dentro de los paréntesis se encontrarán los diferentes argumentos en los que se especificarán sus funciones.
- **GraficasTFG.xlsx**: esta parte de la fórmula alude al libro de Excel donde se encuentran los datos que se quieren utilizar.

- **1_2**: indica el nombre de la hoja en la que se encuentran los datos a calcular dentro del libro indicado como GraficasTFG. En este caso se corresponde a la muestra 2 del usuario número 1.
- **$A$1:$AX$1**: hace referencia al intervalo de celdas de las cuales se quieren calcular el mínimo perteneciente al libro y la hoja especificadas anteriormente. En este caso seleccionamos las primeras 100 muestras.

Al igual que en el cálculo de los máximos, de nuevo, para no tener que escribir la fórmula por cada muestra de cada usuario, se ha desarrollado un programa en C++ para crear el script de los mínimos.

En lo que respecta al programa, lo primero que es necesario, es la creación otra vez de un objeto de tipo *ofstream* que cree un archivo de salida en el que escribir los scripts. Esta vez, dicho objeto se llamará ScriptMIN y el archivo de salida creado ScriptMIN.txt.

```
ofstream escribirMIN("ScriptMIN.txt");
```

Figura 40. Creación del objeto para el fichero de salida.

La siguiente parte del programa será idéntica al cálculo de los máximos, pero con la excepción de poner MIN en vez de MAX, por lo que se han cambiado dichas palabras y ha quedado de la siguiente manera:

```
for(int i=0;i<24;i++)
{

    for (int j=0;j<8;j++)
    {

        escribirMIN<<"=MIN('[GraficasTFG.xlsx]"<<(i+1)<<"_"<<(j+1)<<"'!$A$1:$AX$1);=MIN('[GraficasTFG.xlsx]"<<(i+1)
        escribirMIN<<endl;

    }

    escribirMIN<<endl<<endl;

}
```

Figura 41. Doble bucle *for*.

Más adelante, una vez el archivo ScriptMIN.txt está ya rellenado con todos los scripts, lo único que faltaría será seleccionarlos todos y copiarlos en la hoja de Microsoft Excel donde se están calculando todos los mínimos. El resultado de esta operación quedaría de la siguiente manera:

MINIMOS				
MIN_1	MIN_2	MIN_3	MIN_4	MIN_5
5,90654	5,81648	6,02902	7,03268	6,53174
5,96245	6,05016	6,20848	6,16213	6,12434
6,05164	5,91804	6,69763	6,57691	6,55223
6,06867	5,49614	5,89326	5,84328	5,93156
6,45086	5,74932	6,33156	6,18452	6,26906
6,36938	5,69684	6,20808	6,76414	6,01259
5,78149	5,59086	5,70844	5,22983	5,7463
5,84452	6,10864	5,78241	5,78922	5,40565
5,11739	5,48211	5,16285	5,30574	5,47414
6,32197	6,4445	6,20987	6,47547	6,70019
6,48418	6,30488	6,0146	6,00348	5,88451
6,14773	6,51029	5,08606	6,06004	
5,76771	5,60767	6,23991	5,8458	5,44871
5,30855	4,94498	5,48018	4,84366	5,15947
5,41726	4,71144	4,45321	4,30033	4,04838
6,7006	6,59292	6,69905	6,95597	5,71197

Figura 42. Representación de los mínimos.

Por último, como se especificó en el punto del diseño, el parámetro que realmente interesa a la hora de convertirlo en atributo para su posterior introducción en WEKA, es la media aritmética de los mínimos.

Para ello se volverá a recurrir a la fórmula de Excel llamada Promedio, la cual hará la media aritmética de las celdas establecidas en el argumento de la función. Por ejemplo, en el caso de la siguiente imagen, se calcularía la media aritmética del intervalo de celdas comprendidas entre la C9 y la G9.

Figura 43. Fórmula de Excel para calcular el promedio.

Hay que destacar que, al calcular los mínimos de los datos, como se realiza un muestreo de 100, salen 10 mínimos por muestra de cada usuario, de las cuales sólo valdrán 5 de ellos; el resto se deberán borrar usuario a usuario.

5.3.3 Cálculo de los periodos

En este apartado se va a explicar cómo se ha llevado a cabo el cálculo de los periodos de las gráficas de los datos de todos los usuarios. Para ello, hay que remarcar que se calculará el periodo entre los máximos y entre los mínimos, pero será de manera prácticamente idéntica a todos ellos, por lo que se explicará una vez, señalando las diferencias que pueda haber en cada parte.

En primer lugar, se va a mostrar la fórmula de Excel que se ha escogido para llevar a cabo este cálculo:

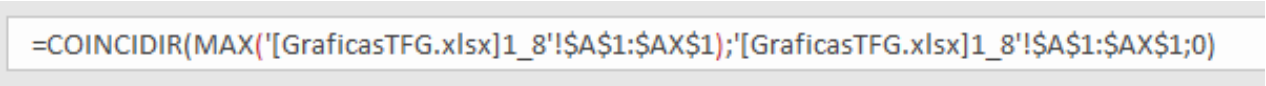

Figura 44. Fórmula de Excel para calcular los periodos.

A continuación, se van a explicar detalladamente cada uno de los diferentes parámetros de los que se compone la fórmula usada:

- **COINCIDIR**: es el nombre de la función de Excel que se va a utilizar. Esta función busca un elemento determinado, en este caso, el máximo o el mínimo en un intervalo para devolver la posición exacta de dicho elemento.
- **MAX**: se refiere al elemento del cual se tiene que buscar la posición. En este caso la imagen hace referencia a un máximo, pero también se podrá utilizar para mínimos poniendo la palabra MIN.
- **GraficasTFG.xlsx**: hace referencia al libro de Excel del cual queremos calcular la posición.
- **1_8**: indica la hoja en la que se encuentran los elementos a calcular del libro anteriormente señalado.
- **$A$1:$AX$1**: hace referencia a las celdas de las cuales se quiere calcular el periodo perteneciente al libro y la hoja detalladas anteriormente. En este caso seleccionamos las primeras 100 muestras.
- **0**: se trata del tipo de coincidencia, es decir, especifica cómo Excel hace coincidir el valor buscado en las celdas seleccionadas. Puede ser -1, 0 y 1. En este caso, se ha elegido el tipo 0 ya que se pretende encontrar el primer valor exactamente igual al valor que se busca.

Una vez más, al igual que en los cálculos de los anteriores parámetros, para no escribir la fórmula previamente descrita para cada muestra de usuario y así reducir el tiempo que tomará

este desarrollo, se ha propuesto un programa en C++ muy similar a los mencionados en los anteriores puntos.

Para empezar, el programa constará de nuevo de un objeto de tipo *ofstream* llamado ScriptT que se utilizará para abrir el archivo o fichero de salida llamado también ScriptT.txt.

```
ofstream escribirT ("ScriptT.txt");
```

Figura 45. Creación del objeto para el fichero de salida.

Por otro lado, la función principal del programa estará formada por un doble bucle *for* que irá aumentando dos variables para obtener el número y muestra del usuario correspondiente en cada caso. La explicación de esto es exactamente igual que en los apartados anteriores.

```
for(int i=0;i<24;i++)
{

    for (int j=0;j<8;j++)
    {

        escribirT<<"=COINCIDIR(MAX('[GraficasTFG.xlsx]"<<(i+1)<<"_"<<(j+1)<<"'!$A$1:$AX$1);
        escribirT<<endl;

    }

    escribirT<<endl<<endl;

}
```

Figura 46. Doble bucle *for*.

Una vez se han copiado todos los scripts a la hoja de Excel donde se quieren calcular los periodos de las muestras, es necesario entender que la función COINCIDIR, ya sea con los máximos o con los mínimos, nos devuelve la posición de dichos máximos o mínimos, pero no el periodo de estos. Es por esto por lo que será necesario calcular la diferencia de estas posiciones devueltas por la función, para así obtener el periodo.

	T1	T2	T3	T4	T5	(T2-T1)	(T3-T2)	(T4-T3)	(T5-T4)
1_1	111	317	531	731	945	206	214	200	214
1_2	70	263	465	664	860	193	202	199	196
1_3	117	313	506	701	896	196	193	195	195
1_4	191	386	580	773	970	195	194	193	197
1_5	72	263	449	636	838	191	186	187	202
1_6	90	291	487	688	885	201	196	201	197

Figura 47. Representación de los periodos.

Para finalizar hay que destacar, al igual que en anteriores casos, que el parámetro que realmente interesa es la media aritmética de cada periodo, por lo que usando una vez más la función de Excel llamada Promedio, se obtendrá el parámetro deseado.

5.3.4 Cálculo del resto de los parámetros

Para terminar con el capítulo del cálculo de parámetros, se va a indicar como se han calculado el resto de los parámetros expuestos en el apartado del diseño de la manera más sencilla y breve posible, ya que no requiere de especial dificultad.

Los parámetros que se desean calcular se explican a continuación y se pueden observar en la siguiente tabla:

Tabla 3. Representación del resto de parámetros.

MediaMAX_1/MIN	MediaMAX_2/MIN	MediaMAX_1/MAX_2	USUARIO
2,841329609	2,589245294	1,10079745	1
2,864293463	2,813163965	1,019738592	1
2,975986842	2,63015324	1,133378025	1
2,871866298	3,01068989	0,957065195	1
2,88633385	2,866672648	1,006200572	1
2,806861721	2,780145684	1,010450553	1
3,219330895	3,023230496	1,06722189	1
3,267648366	3,148927174	1,039524238	1
3,281778252	3,123438661	1,052706417	2

- **MediaMAX_1/MIN**: este parámetro indica la división entre el promedio calculado del máximo ‘1’ de una muestra entre el promedio del mínimo de dicha misma muestra. Esta operación se llevará a cabo mediante la opción de la división de Excel: C9/J9, donde C9 se corresponde en este ejemplo al máximo ‘1’ y J9 al mínimo.

- **MediaMAX_2/MIN**: este parámetro indica la división entre el promedio calculado del máximo 2 de una muestra entre el promedio del mínimo de dicha misma muestra. La forma de obtenerlo es exactamente la misma que el parámetro anterior.
- **MediaMAX_1/MAX_2**: este parámetro indica la división entre el promedio calculado del máximo 1 de una muestra entre el promedio del máximo 1 de dicha misma muestra. La operación necesaria volverá a ser la misma que en casos anteriores.
- **USUARIO**: hace referencia al usuario del cual se están calculando los parámetros. Este parámetro servirá más tarde para el reconocimiento en WEKA, siendo este el atributo más importante.

5.4 Preparación y clasificación de los datos

El objetivo de este capítulo resolverá la problemática a la hora de meter los datos con todos sus parámetros calculados en WEKA para poder trabajar en dicha plataforma. Antes de eso es necesario un paso intermedio entre la hoja de Excel donde se encuentran nuestros datos y el archivo con extensión .arff que reconocerá WEKA.

5.4.1 Creación del archivo .arff

Para empezar, será imprescindible convertir nuestros datos de la tabla de Excel en un archivo de texto donde posteriormente se modificará y para ello es necesario guardar dicha hoja Excel como extensión .csv en las opciones de guardado.

Una vez se ha guardado el archivo, éste se abrirá con un editor de texto que en este caso será Notepad ++, ya que nos permite funciones que serán de utilidad más tarde. El editor se mostrará de la siguiente manera y aunque parezca un poco desordenado, no tiene ninguna trascendencia, ya que el siguiente paso será modificarlo y dar entonces el formato .arff buscado descrito previamente.

```
1_1 17,71082    16,10068    6,263292    2,841329609 2,589245294 1,1007974   208,5   207,25  212,25  1
1_2 17,47738    17,15588    6,101512    2,864293463 2,813163965 1,019738592 197,5   197,75  199 1
1_3 18,8476 16,66188    6,35929 2,975986842 2,63015324  1,1333780252    194,75  195,5   196,5   1
```

Figura 48. Primera vista del fichero .csv.

En primer lugar, se borrarán todos los nombres de usuario con su muestra, por ejemplo 1_1, 1_2…. ,ya que la única finalidad de ello era tener un mayor control en el Excel, pero ahora la tarea prioritaria es centrarse en dejar únicamente los atributos futuros.

A continuación, se sustituirán las comas de los decimales por puntos, ya que el archivo deseado .arff solo admite el punto como indicador de un decimal. Por otro lado, también será necesario sustituir los espacios entre un dato y otro por comas, indicando así la separación de estos.

Estos reemplazos se realizarán con la función de Notepad++ reemplazar, donde se introduce un texto a sustituir por otro y se puede observar en la siguiente imagen:

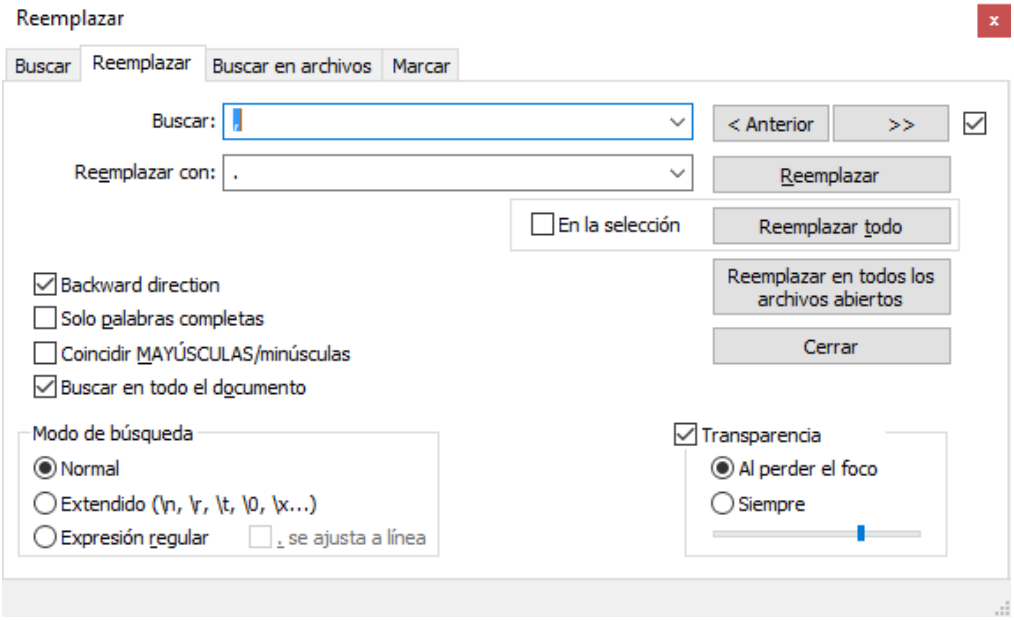

Figura 49. Ventana de reemplazar en Notepad++.

Finalmente, los datos quedarán de la siguiente manera ya con el formato deseado e irán precedidos de la palabra @data:

```
17  @data
18
19  17.71082,16.10068,6.263292,2.841329609,2.589245294,1.10079745,208.5,207.25,212.25,1
20  17.47738,17.15588,6.101512,2.864293463,2.813163965,1.019738592,197.5,197.75,199,1
21  18.8476,16.66188,6.35929,2.975986842,2.63015324,1.133378025,194.75,195.5,196.5,1
```

Figura 50. Visualización final de los datos en el archivo .arff.

El próximo paso a seguir es hacer los cambios oportunos para dar formato arff a dicho archivo de texto. Para ello, hay que tener claro que nuestros datos irán divididos en dos bases de datos o dos ficheros diferentes como se explicó en el punto del diseño, uno dedicado al entrenamiento de los datos, los cuales tendrán los datos correspondientes al 70 % de los usuarios y el otro para probar o testear los datos con el porcentaje restante de los usuarios

Por tanto, en el encabezado de nuestras bases de datos será necesaria la inclusión del nombre de cada una de ellas gracias a la palabra @relation, quedando de tal manera:

```
1
2   @relation BDtest
3
```

```
1
2   @relation BDtrain
3
```

Figura 51. Visualización del encabezado del fichero .arff.

Mas adelante, como último paso, será necesario indicar los atributos. Es muy importante que se introduzcan en el mismo orden en las dos bases de datos, así como en el mismo orden en el que se encuentran los datos. Quedarán de la siguiente manera:

```
4  @attribute MediaMAX_1 numeric
5  @attribute MediaMAX_2 numeric
6  @attribute MediaMIN numeric
7  @attribute MediaMAX_1/MIN numeric
8  @attribute MediaMAX_2/MIN numeric
9  @attribute MediaMAX_1/MAX_2 numeric
10 @attribute MEDIA_T_MAX_1 numeric
11 @attribute MEDIA_T_MAX_2 numeric
12 @attribute MEDIA_T_MIN numeric
13
14 @attribute USUARIO {'1','2','3','4','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24'}
15
```

Figura 52. Visualización de los atributos del fichero .arff.

Para finalizar, hay que destacar como se puede apreciar en la imagen superior, que el último atributo, será el atributo según el cual queremos clasificar nuestras instancias. Una vez tenemos declarados ambas bases de datos en el formato correcto, se guardarán como .arff en vez de .txt y se procederá a introducirlos en WEKA.

5.4.2 Introducción y clasificación de los datos en WEKA

Para finalizar este capítulo, se va a proceder a la explicación de cómo se ha desarrollado la última parte de este trabajo, que consistirá en la introducción de los datos y su posterior clasificación en el programa WEKA.

Una vez se tienen diferenciados los dos ficheros a usar, lo primero es introducir la base de datos de entrenamiento en la pestaña *preprocess* del programa. Para ello, se seleccionará la opción *open file* y, acto seguido, el fichero mencionado.

Al introducir dicho fichero en WEKA se puede observar cómo aparecen los atributos representados en la interfaz del programa. Esto quiere decir que no ha habido ningún problema

y WEKA ha reconocido perfectamente el archivo. La interfaz lucirá de manera similar a la siguiente imagen:

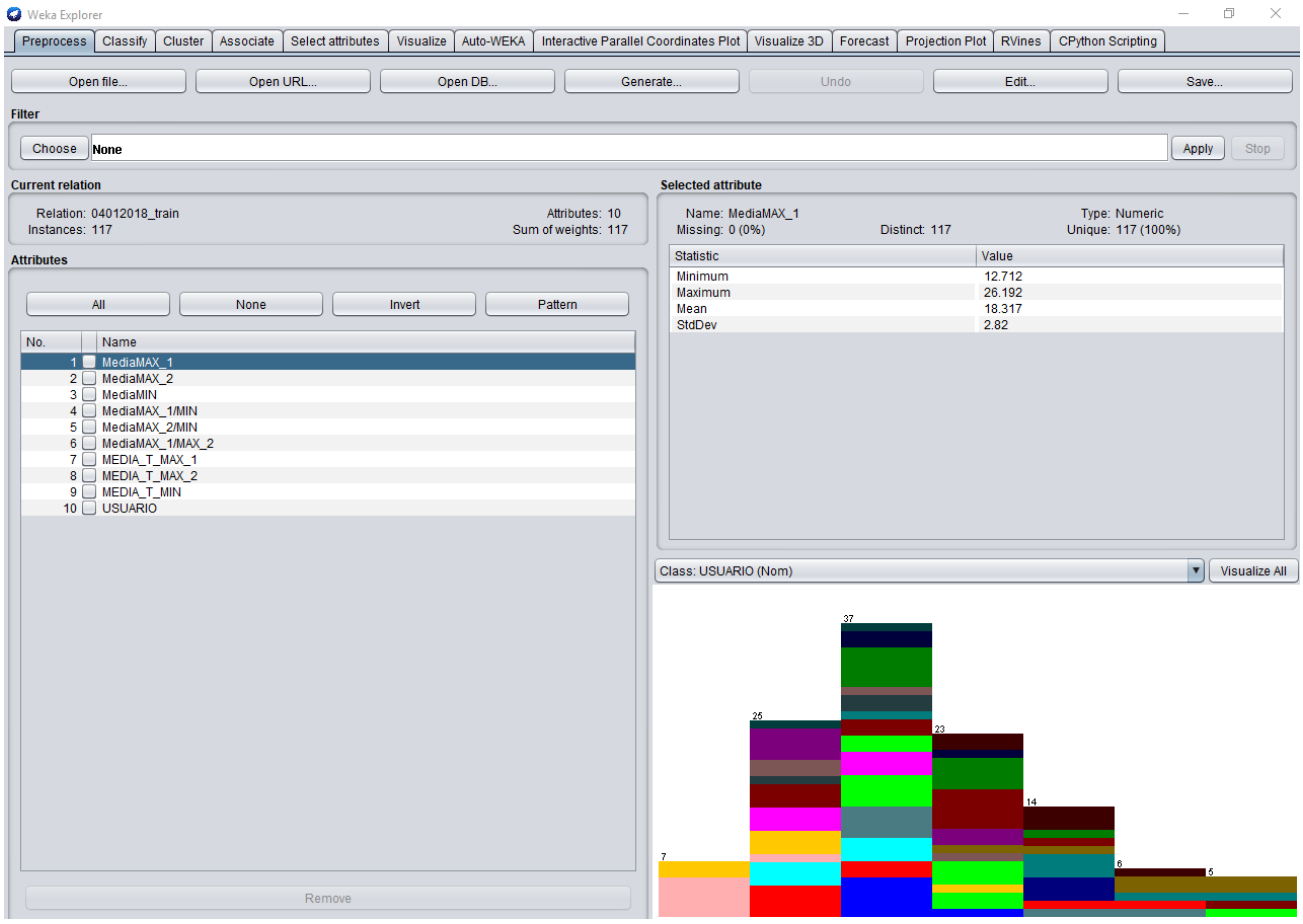

Figura 53. Interfaz principal de la ventana preprocess en WEKA.

Mas adelante, una vez en la interfaz de *classify* es importante seleccionar la opción con la que se desean tratar los datos y elegir la pestaña *supplied test set* como se ha explicado anteriormente. En este punto se procederá a importar la base de datos de prueba.

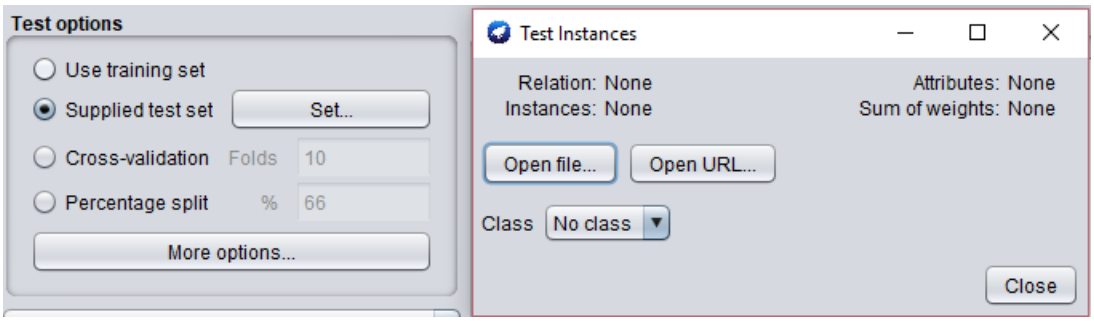

Figura 54. Ventana *test options*.

A continuación, el siguiente paso será proceder a la clasificación de los datos mediante la elección de los clasificadores que se especificaron en el apartado del diseño de la solución. Para llevar a cabo esta operación, se seleccionará la pestaña *choose* y se escogerán dichos clasificadores.

Habrá, por tanto, dos evaluaciones y sus respectivos resultados serán totalmente distintos:

1- **Elección de NaivesBayes**.

   En el caso de elegir este clasificador, basado en algoritmos bayesianos, se pueden observar los siguientes resultados en la ventana *output*.

   Por un lado, se observan los porcentajes de acierto en un resumen de la clasificación en la que se observa que un 51.2821% de las instancias han sido correctamente clasificadas.

```
Correctly Classified Instances          20               51.2821 %
Incorrectly Classified Instances        19               48.7179 %
Kappa statistic                          0.4879
Mean absolute error                      0.0437
Root mean squared error                  0.1852
Relative absolute error                 52.7335 %
Root relative squared error             90.9744 %
Total Number of Instances               39
```

Figura 55. Visualización de los porcentajes de aciertos.

   Si se desea saber cuáles han sido exactamente los errores de clasificación, basta con echar un vistazo a la matriz de confusión, donde se pueden observar fuera de la diagonal principal las instancias mal clasificadas.

```
a b c d e f g h i j k l m n o p q r s t u v w   <-- classified as
0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = 1
0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 | b = 2
0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = 3
0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = 4
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | e = 6
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | f = 7
0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = 8
0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = 9
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = 10
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = 11
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 | k = 12
0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 | l = 13
0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 | m = 14
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 | n = 15
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 | o = 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 | p = 17
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 | q = 18
0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 | r = 19
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 | s = 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 | t = 21
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | u = 22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 | v = 23
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 | w = 24
```

Figura 56. Visualización de la matriz de confusión para NaivesBayes.

2- **Elección de Random Forest**.

En el caso de haber escogido este clasificador, se pueden observar los siguientes resultados.

Esta vez el porcentaje de instancias correctamente clasificadas que aparece en el resumen de la clasificación asciende a un 53.8462%.

```
Correctly Classified Instances          21               53.8462 %
Incorrectly Classified Instances        18               46.1538 %
Kappa statistic                          0.5145
Mean absolute error                      0.0581
Root mean squared error                  0.1661
Relative absolute error                 70.0269 %
Root relative squared error             81.5645 %
Total Number of Instances               39
```

Figura 57. Visualización del porcentaje de aciertos.

Por último, para observar de nuevo las instancias mal clasificadas, se acudirá a la matriz de confusión:

```
a b c d e f g h i j k l m n o p q r s t u v w   <-- classified as
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 | a = 1
0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 | b = 2
0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | c = 3
0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | d = 4
0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | e = 6
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | f = 7
0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | g = 8
0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | h = 9
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | i = 10
1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | j = 11
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 | k = 12
0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 | l = 13
0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 | m = 14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 | n = 15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | o = 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 | p = 17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 | q = 18
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 | r = 19
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 | s = 20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 | t = 21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 | u = 22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 | v = 23
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 | w = 24
```

Figura 58. Visualización de la matriz de confusión para Random Forest.

Hay que destacar que se han probado otros clasificadores como RandomTree, J48 o IB1, los cuales también podrían haber sido aceptables para los datos, pero los porcentajes son más bajos y no se han incluido en la memoria, decidiendo así exponer sólo los dos más representativos de todos ellos.

6 Problemas surgidos

En este punto del trabajo, se va a proceder a explicar los problemas que han surgido a lo largo del desarrollo del mismo. Todos estos problemas van a estar relacionados con el tratamiento y el preprocesamiento de los datos antes de su introducción en WEKA, ya que en la parte de clasificación con los algoritmos de machine learning no se han encontrado problemas destacables.

El primer problema que surge durante la realización de este trabajo es la indecisión a la hora de tomar un número determinado de muestras de cada usuario para calcular la magnitud. Como se ha explicado anteriormente, la primera idea fue tomar las primeras 1000 muestras de cada magnitud, pero los datos eran irregulares y no era posible obtener características de ellos, por lo que se decidió tomar las 5000 primeras muestras de cada usuario y proceder a calcular su magnitud.

A la hora de representar las 5000 muestras, las gráficas quedaban ilegibles al tener un periodo muy pequeño, haciendo su estudio muy complicado. Esto se convierte en un gran inconveniente, ya que su estudio es necesario para la posterior elección de los parámetros característicos de cada usuario.

Debido a esto, se decidió escoger las ultimas 1000 muestras de las 5000 previamente obtenidas, para que así, los datos fueran representados de una forma más reconocible y legible para la vista.

Una vez se ha decidido escoger los máximos absolutos (máximos 1), los segundos máximos (máximos 2) y los mínimos absolutos, el siguiente problema surgirá a la hora de establecer el intervalo de celdas sobre el cual se calcularán estos parámetros característicos.

En un principio se pensó establecer un rango de 50 magnitudes y obtener a partir de ellos los parámetros mencionados, pero eran demasiados datos y era necesario la filtración manual de muchos de ellos. Por esta razón se acabó por desechar esa opción y se optó por establecer un rango de 100 magnitudes.

De esta manera, ampliando el abanico de celdas en la operación, no se perdían muestras válidas y, además, el filtrado manual requerido para borrar las muestras incorrectas era considerablemente menor.

Por último, surgió otro problema a la hora de seleccionar más parámetros para así tener un mayor número de atributos y proporcionar más información a los clasificadores de WEKA. En este caso se pensó en establecer un atributo que fuera la suma de los máximos '1', los máximos '2' y la suma de los mínimos de cada usuario.

Al llevar a cabo esta operación e introducirlo en nuestras bases de datos en WEKA, se pudo observar que, tanto el número de instancias correctamente clasificadas como su porcentaje correspondiente bajaba ligeramente, por lo que se decidió no incluir dichos atributos en las tablas.

7 Conclusiones

El desarrollo del último capítulo de este Trabajo de Fin de Grado va a tratar acerca de dos puntos bien diferenciados. Por un lado, el cumplimiento de los objetivos establecidos al principio del documento y por otro lado el resumen de lo que se podrían ser nuevas líneas de trabajo futuro e investigación acerca del tema tratado a lo largo del trabajo.

A nivel personal, la elección de este trabajo, que quizás se salga un poco de las líneas tradicionales de la electrónica, era principalmente conocer otros campos de la ingeniería como es el área del aprendizaje automático, del que el alumno partía casi del total desconocimiento. Una vez acabado el trabajo se puede afirmar que se han adquirido numerosos conceptos y técnicas desconocidos hasta el momento, así como el interés en campos de la ingeniería que se creían más lejanos. Todo esto podría facilitar en un futuro seguir tratando y estudiando sobre este tema.

Este trabajo, también ha servido para afianzar y extender algunos conocimientos que se han estudiado a lo largo del grado como la programación en distintos lenguajes o el manejo de ciertos programas de cálculo.

Por tanto, se puede concluir con la seguridad de haber adquirido grandes conocimientos que sin duda servirán para ponerlos en práctica en un futuro, ya que han resultado de gran interés y utilidad en muchos escenarios diferentes.

7.1 Objetivos cumplidos

Al principio de este documento, se establecieron una serie de objetivos con el fin de analizar al final del trabajo, concretamente en este punto, si han sido o no alcanzados. Además, se puede anticipar, que en la mayoría de ellos el resultado ha sido satisfactorio.

Por tanto, a continuación, se van a presentar dichos objetivos, ofreciendo una evaluación sobre ellos y explicando el porqué de su alcance de forma positiva o negativa.

- **Preparación y procesamiento de las bases de datos incluyendo el uso de los lenguajes de programación necesarios.**

  El primer objetivo establecido a valorar se puede juzgar de manera positiva unilateralmente, ya que gran parte del trabajo se ha desarrollado trabajando con estas bases de datos, incluyendo la realización de programas para diferentes cálculos en Java y C++.

- **Uso de clasificadores, así como su posterior análisis y su impacto en las muestras de la base de datos.**

  Este segundo objetivo del trabajo también se puede fijar como alcanzado favorablemente, ya que se han elegido los mejores clasificadores para nuestros datos y se ha explicado y analizado su impacto en los resultados.

- **Usar un programa de minería de datos, en este caso WEKA, y entender sus funcionalidades.**

  WEKA tiene infinidad de funcionalidades, pero a pesar de eso, este punto se puede marcar también como conseguido de forma favorable, ya que el alumno ha corroborado y explicado las distintas opciones que han sido usadas, ofrecidas por el programa a lo largo del trabajo, sin las cuales hubiera sido imposible la realización de éste.

- **Conseguir un porcentaje de acierto en el reconocimiento de usuarios notablemente alto.**

  Este último objetivo, y quizás uno de los más importantes, ya que es en el que se basa el fin principal del trabajo, se podría decir que se ha conseguido alcanzar a medias.

  Por una parte, es cierto que el porcentaje de acierto en el reconocimiento de las instancias ha sido superior al 50 % pero, por otra parte, este porcentaje no es lo suficientemente elevado como para declarar este resultado como satisfactorio.

  En el siguiente punto se expondrán algunas de las ideas sobres las cuales sería interesante trabajar en un futuro de acuerdo a subir el porcentaje de reconocimiento referido en este punto.

7.2 Trabajos futuros

Para terminar este trabajo, se pretenden establecer algunas ideas del alumno que puedan servir como líneas futuras de trabajo e investigación para desarrollar más el reconocimiento de usuario a través de la pisada, ya que aunque se han cumplido satisfactoriamente la mayor parte

de los objetivos, siempre es posible una mejora del estudio. De esta manera se podría conseguir ampliar el alcance de algunos objetivos propuestos como el porcentaje de reconocimiento.

En el caso de este trabajo, como se ha explicado a lo largo de él, la base de datos con la recogida de mediciones de las pisadas de los usuarios ha sido dada al alumno. Se propone entonces, en un futuro, otra manera de recoger esta base de datos empleando otros dispositivos como giroscopios que ayuden a tener mejores y más claras mediciones en los datos.

Por último, también sería interesante añadir más atributos a las bases de datos que se introducen en WEKA con el fin de dar más información a los clasificadores y subir el porcentaje de acierto. Para ello habría que diseñar y calcular nuevos parámetros a partir de relaciones entre las características propias de cada usuario.

Bibliografía

[1] Wikipedia. Biometría. https://es.wikipedia.org/wiki/Biometr%C3%ADa#Historia. Consultado en diciembre de 2018.

[2] History of biometrics https://www.biometricupdate.com/201802/history-of-biometrics-2. Consultado en diciembre de 2018.

[3] Raúl Sánchez Reíllo, Identificación Biométrica y su unión con las Tarjetas Inteligentes, Ágora sic, 2000. https://revistasic.com/revista39/pdf_39/SIC_39_agora.PDF. Consultado en diciembre de 2018.

[4] Ventajas y desventajas de los diversos tipos de sistemas biométricos. http://dchain.com/ventajas-y-desventajas-de-los-diversos-tipos-de-sistemas-biometricos/ Consultado en diciembre de 2018.

[5] Conferencia Española de Biometría, X Conferencia Española de Biometría: Libro de resúmenes, 2005. https://books.google.es/books?id=UuMsuoEj-JQC&printsec=frontcover&dq=X+conferencia+espa%C3%B1ola+de+biometria.+libro+de+resumenes&hl=es&sa=X&ved=0ahUKEwj939Dd3dPcAhWMa8AKHcObBl4Q6AEIJzAA#v=onepage&q=X%20conferencia%20espa%C3%B1ola%20de%20biometria.%20libro%20de%20resumenes&f=false Consultado en diciembre de 2018.

[6] Jesús Cámara, Análisis de la marcha: sus fases y variables espaciotemporales, Entramado vol. 7, 2011. https://www.redalyc.org/articulo.oa?id=265420116010 Consultado en diciembre de 2018.

[7] Adriana Isabel Agudelo Mendoza, Tatiana Julieth Briñez Santamaria, Vanessa Guarín Urrego, Juan Pablo Ruiz Restrepo, Marlly Carolina Zapata García, Marcha: descripción, métodos, herramientas de evaluación y parámetros de normalidad reportados en la literatura, CES Movimiento y Salud, 2013. https://docplayer.es/21277429-Marcha-descripcion-metodos-herramientas-de-evaluacion-y-parametros-de-normalidad-reportados-en-la-literatura.html . Consultado en diciembre de 2018.

[8] Wikipedia, Análisis de la marcha. https://es.wikipedia.org/wiki/An%C3%A1lisis_de_la_marcha#T%C3%A9cnicas Consultado en diciembre de 2018.

[9] Elibe Frank, Mark A- Hall, Ian H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016. https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf Consultado en enero de 2018.

[10] Wikipedia, Aprendizaje automático, https://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico) Consultado en enero de 2018.

[11] Francisco Navas Moreno, Introducción a la minería de datos con WEKA: aplicación a un problema económico. http://tauja.ujaen.es/bitstream/10953.1/6984/1/TFG%20-%20navas%20moreno%2C%20%20Francisco.pdf?_sm_byp=iVVrjrTn86vH5rts Consultado en enero de 2018.

[12] Diego García Morate, Manual de WEKA, 2004 https://docplayer.es/146074-Manual-de-weka-diego-garcia-morate-diego-garcia-morate-at-gmail-com.html Consultado en enero de 2018.

Anexo A: Planificación y Presupuesto

A continuación, se va a llevar a cabo un desglose de las tareas que se han realizado a lo largo de este trabajo fin de grado, lo que facilitará posteriormente un cálculo aproximado sobre su coste.

A.1 Planificación

Debido a la complejidad de un trabajo de estas características se ha optado por dividirlo en distintas fases, las cuales se van a comentar a continuación:

Fase 1: Documentación inicial

I. Estudio de la plataforma WEKA, C++ y Java (45 horas)
II. Preparación de las herramientas de trabajo (10 horas)
III. Búsqueda y realización de tutoriales y aplicaciones sencillas. (25 horas)

Fase 2: Desarrollo del trabajo

I. Preparación y preprocesamiento de las bases de datos (60 horas)
II. Representación de los datos (30 horas)
III. Cálculo de los parámetros de los datos (30 horas)
IV. Clasificación de los datos (20 horas)

Fase 3: Elaboración de la memoria

I. Redacción de la memoria (60 horas)
II. Corrección y maquetación (20 horas)

Tabla 4 - Desglose de tareas

FASES	HORAS EMPLEADAS
Documentación inicial	80
Desarrollo de la aplicación	140
Elaboración de la memoria	80
TOTAL	**300**

A.2 Presupuesto del Trabajo Fin de Grado

A.2.1 Costes materiales

Los costes materiales para el desarrollo de este trabajo ha sido un ordenador de altas prestaciones con el cuál poder trabajar y procesar todos los datos.

Se ha considerado un periodo de amortización de 3 años y el tiempo en el que ha sido usado para el desarrollo, de aproximadamente 6 meses.

Tabla 5. Costes materiales.

CONCEPTO	PRECIO (€)
Ordenador Altas prestaciones	116.7
TOTAL	**116.7**

A.2.2 Costes de personal

Para la realización de este trabajo, ha sido necesaria la presencia de un jefe de proyecto y un ingeniero.

Tabla 6 – Costes de Personal

OCUPACIÓN	HORAS	PRECIO/HORA	IMPORTE (€)
Jefe de proyecto	30	90	2700
Ingeniero	270	60	16200
TOTAL	**300**		**18900**

A.2.3 Costes totales

Tabla 7 – Costes Totales

CONCEPTO	PRECIO (€)
Costes de personal	18900
Costes materiales	116.7
Costes indirectos (20%)	3803.4
Subtotal	22820
IVA (18%)	4107.6
TOTAL	**26927.6**

El coste total del proyecto es de VEINTISÉIS MIL NOVECIENTOS VEINTISIETE EUROS CON SESENTA CÉNTIMOS.